

# RL78/L13

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>).

## 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.  
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.  

Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

## NOTES FOR CMOS DEVICES

- (1) **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 VIL (MAX) and VIH (MIN) due to noise, etc., 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 VIL (MAX) and VIH (MIN).
- (2) **HANDLING OF UNUSED INPUT PINS:** Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.
- (3) **PRECAUTION AGAINST ESD:** A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that 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 should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.
- (4) **STATUS BEFORE INITIALIZATION:** Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.
- (5) **POWER ON/OFF SEQUENCE:** In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.
- (6) **INPUT OF SIGNAL DURING POWER OFF STATE :** Do not input signals or an I/O pull-up power supply while the device is not powered. 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. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

# How to Use This Manual

## Readers

This manual is intended for user engineers who wish to understand the functions of the RL78/L13 and design and develop application systems and programs for these devices. The target products are as follows.

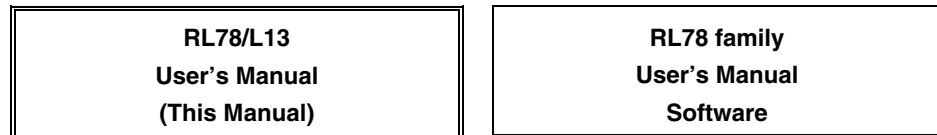
- 64-pin: R5F10WLx (x = A, C-G)
- 80-pin: R5F10WMx (x = A, C-G)

## Purpose

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

## Organization

The RL78/L13 manual is separated into two parts: this manual and the software edition (common to the RL78 family).



- 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/L13 Microcontroller instructions:
  - Refer to the separate document **RL78 family Software User's Manual (R01US0015E)**.



<b>Conventions</b>	Data significance:	Higher digits on the left and lower digits on the right
	Active low representations:	$\overline{xxx}$ (overscore over pin and signal name)
	<b>Note:</b>	Footnote for item marked with <b>Note</b> in the text
	<b>Caution:</b>	Information requiring particular attention
	<b>Remark:</b>	Supplementary information
	Numerical representations:	Binary            ...xxxx or xxxxB Decimal            ...xxxx Hexadecimal        ...xxxxH

**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/L13 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
Semiconductor Reliability Handbook	R51ZZ0001E

**Note** See the "Semiconductor Package Mount Manual" website (<http://www.renesas.com/products/package/manual/index.jsp>).

**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 <sup>®</sup> technology licensed from Silicon Storage Technology, Inc.
--

# CONTENTS

<b>CHAPTER 1 OUTLINE.....</b>	<b>1</b>
<b>1.1 Features.....</b>	<b>1</b>
<b>1.2 List of Part Numbers .....</b>	<b>4</b>
<b>1.3 Pin Configuration (Top View) .....</b>	<b>6</b>
1.3.1 64-pin products.....	6
1.3.2 80-pin products.....	7
<b>1.4 Pin Identification.....</b>	<b>8</b>
<b>1.5 Block Diagram .....</b>	<b>9</b>
1.5.1 64-pin products.....	9
1.5.2 80-pin products.....	10
<b>1.6 Outline of Functions.....</b>	<b>11</b>
<b>CHAPTER 2 PIN FUNCTIONS .....</b>	<b>13</b>
<b>2.1 Port Function .....</b>	<b>13</b>
2.1.1 64-pin products.....	14
2.1.2 80-pin products.....	17
<b>2.2 Functions Other than Port Pins .....</b>	<b>20</b>
2.2.1 With functions for each product.....	20
2.2.2 Description of Functions .....	22
<b>2.3 Connection of Unused Pins .....</b>	<b>24</b>
<b>2.4 Block Diagrams of Pins .....</b>	<b>27</b>
<b>CHAPTER 3 CPU ARCHITECTURE .....</b>	<b>43</b>
<b>3.1 Memory Space .....</b>	<b>43</b>
3.1.1 Internal program memory space.....	52
3.1.2 Mirror area.....	55
3.1.3 Internal data memory space.....	57
3.1.4 Special function register (SFR) area .....	57
3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area .....	57
3.1.6 Data memory addressing .....	58
<b>3.2 Processor Registers.....</b>	<b>60</b>
3.2.1 Control registers .....	60
3.2.2 General-purpose registers.....	62
3.2.3 ES and CS registers.....	63
3.2.4 Special function registers (SFRs) .....	64
3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers) .....	70
<b>3.3 Instruction Address Addressing.....</b>	<b>80</b>

3.3.1	Relative addressing .....	80
3.3.2	Immediate addressing .....	80
3.3.3	Table indirect addressing .....	81
3.3.4	Register direct addressing.....	82
<b>3.4</b>	<b>Addressing for Processing Data Addresses .....</b>	<b>83</b>
3.4.1	Implied addressing .....	83
3.4.2	Register addressing .....	83
3.4.3	Direct addressing .....	84
3.4.4	Short direct addressing .....	85
3.4.5	SFR addressing.....	86
3.4.6	Register indirect addressing.....	87
3.4.7	Based addressing.....	88
3.4.8	Based indexed addressing .....	92
3.4.9	Stack addressing.....	93
<b>CHAPTER 4</b>	<b>PORT FUNCTIONS .....</b>	<b>97</b>
<b>4.1</b>	<b>Port Functions .....</b>	<b>97</b>
<b>4.2</b>	<b>Port Configuration.....</b>	<b>97</b>
4.2.1	Port 0.....	98
4.2.2	Port 1.....	98
4.2.3	Port 2.....	99
4.2.4	Port 3.....	99
4.2.5	Port 4.....	100
4.2.6	Port 5.....	100
4.2.7	Port 6.....	100
4.2.8	Port 7.....	101
4.2.9	Port 12.....	101
4.2.10	Port 13.....	101
<b>4.3</b>	<b>Registers Controlling Port Function .....</b>	<b>102</b>
4.3.1	Port mode registers (PMxx).....	106
4.3.2	Port registers (Pxx).....	107
4.3.3	Pull-up resistor option registers (PUxx) .....	108
4.3.4	Port input mode registers (PIMxx) .....	109
4.3.5	Port output mode registers (POMxx) .....	110
4.3.6	Port mode control registers (PMCxx) .....	111
4.3.7	A/D port configuration register (ADPC) .....	112
4.3.8	Peripheral I/O redirection register (PIOR).....	113
4.3.9	LCD port function registers 0 to 6 (PFSEG0 to PFSEG6) .....	114
4.3.10	LCD input switch control register (ISCLCD) .....	117
<b>4.4</b>	<b>Port Function Operations .....</b>	<b>118</b>
4.4.1	Writing to I/O port .....	118

4.4.2	Reading from I/O port.....	118
4.4.3	Operations on I/O port.....	118
4.4.4	Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers .....	119
<b>4.5</b>	<b>Register Settings When Using Alternate Function .....</b>	<b>121</b>
4.5.1	Basic concept when using alternate function.....	121
4.5.2	Register settings for alternate function whose output function is not used .....	122
4.5.3	Register setting examples for used port and alternate functions.....	124
4.5.4	Operation of ports that alternately function as SEGxx pins .....	138
4.5.5	Operation of ports that alternately function as VL3, CAPL, and CAPH pins.....	140
4.5.6	P45 (port that serves as an analog input pin (IVREF0)) .....	142
<b>4.6</b>	<b>Cautions When Using Port Function.....</b>	<b>143</b>
4.6.1	Cautions on 1-bit manipulation instruction for port register n (Pn).....	143
4.6.2	Notes on specifying the pin settings .....	144
<b>CHAPTER 5</b>	<b>CLOCK GENERATOR .....</b>	<b>145</b>
<b>5.1</b>	<b>Functions of Clock Generator.....</b>	<b>145</b>
<b>5.2</b>	<b>Configuration of Clock Generator .....</b>	<b>147</b>
<b>5.3</b>	<b>Registers Controlling Clock Generator.....</b>	<b>149</b>
5.3.1	Clock operation mode control register (CMC) .....	149
5.3.2	System clock control register (CKC).....	152
5.3.3	Clock operation status control register (CSC) .....	154
5.3.4	Oscillation stabilization time counter status register (OSTC).....	155
5.3.5	Oscillation stabilization time select register (OSTS) .....	157
5.3.6	Peripheral enable registers 0 and 1 (PER0, PER1).....	159
5.3.7	Subsystem clock supply mode control register (OSMC) .....	162
5.3.8	High-speed on-chip oscillator frequency select register (HOCODIV) .....	163
5.3.9	High-speed on-chip oscillator trimming register (HIOTRM) .....	164
<b>5.4</b>	<b>System Clock Oscillator .....</b>	<b>165</b>
5.4.1	X1 oscillator.....	165
5.4.2	XT1 oscillator.....	165
5.4.3	High-speed on-chip oscillator .....	169
5.4.4	Low-speed on-chip oscillator .....	169
<b>5.5</b>	<b>Clock Generator Operation .....</b>	<b>170</b>
<b>5.6</b>	<b>Controlling Clock.....</b>	<b>172</b>
5.6.1	Example of setting high-speed on-chip oscillator .....	172
5.6.2	Example of setting X1 oscillation clock.....	174
5.6.3	Example of setting XT1 oscillation clock .....	175
5.6.4	CPU clock status transition diagram.....	176
5.6.5	Conditions before changing the CPU clock and processing after changing CPU clock.....	182
5.6.6	Time required for switching CPU clock and system clock .....	184
5.6.7	Conditions before stopping clock oscillation .....	185

<b>5.7 Resonator and Oscillator Constants</b> .....	<b>186</b>
<b>CHAPTER 6 TIMER ARRAY UNIT</b> .....	<b>187</b>
<b>6.1 Functions of Timer Array Unit</b> .....	<b>188</b>
6.1.1 Independent channel operation function .....	188
6.1.2 Simultaneous channel operation function.....	189
6.1.3 8-bit timer operation function (channels 1 and 3 only).....	190
6.1.4 LIN-bus supporting function (channel 7 only) .....	191
<b>6.2 Configuration of Timer Array Unit</b> .....	<b>192</b>
6.2.1 Timer count register mn (TCRmn).....	198
6.2.2 Timer data register mn (TDRmn).....	199
<b>6.3 Registers Controlling Timer Array Unit</b> .....	<b>200</b>
6.3.1 Peripheral enable register 0 (PER0).....	201
6.3.2 Timer clock select register m (TPSm) .....	202
6.3.3 Timer mode register mn (TMRmn) .....	205
6.3.4 Timer status register mn (TSRmn) .....	210
6.3.5 Timer channel enable status register m (TEm).....	211
6.3.6 Timer channel start register m (TSm).....	212
6.3.7 Timer channel stop register m (TTm) .....	213
6.3.8 Timer input select register 0 (TIS0) .....	214
6.3.9 Timer output select register (TOS) .....	214
6.3.10 Timer output enable register m (TOEm).....	215
6.3.11 Timer output register m (TOm) .....	216
6.3.12 Timer output level register m (TOLm).....	217
6.3.13 Timer output mode register m (TOMm) .....	218
6.3.14 Input switch control register (ISC) .....	219
6.3.15 Noise filter enable register 1 (NFEN1).....	220
6.3.16 Registers controlling port functions of pins to be used for timer I/O .....	222
<b>6.4 Basic Rules of Timer Array Unit</b> .....	<b>223</b>
6.4.1 Basic rules of simultaneous channel operation function.....	223
6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only) .....	225
<b>6.5 Operation of Counter</b> .....	<b>226</b>
6.5.1 Count clock (fTCLK) .....	226
6.5.2 Start timing of counter .....	228
6.5.3 Operation of counter.....	229
<b>6.6 Channel Output (TOmn pin) Control</b> .....	<b>234</b>
6.6.1 TOmn pin output circuit configuration.....	234
6.6.2 TOmn pin output setting .....	235
6.6.3 Cautions on channel output operation.....	236
6.6.4 Collective manipulation of TOmn bit.....	241
6.6.5 Timer interrupt and TOmn pin output at operation start.....	242

<b>6.7 Timer Input (Tlmn) Control .....</b>	<b>243</b>
6.7.1 Tlmn input circuit configuration.....	243
6.7.2 Noise filter .....	243
6.7.3 Cautions on channel input operation .....	244
<b>6.8 Independent Channel Operation Function of Timer Array Unit.....</b>	<b>245</b>
6.8.1 Operation as interval timer/square wave output .....	245
6.8.2 Operation as external event counter .....	251
6.8.3 Operation as input pulse interval measurement .....	256
6.8.4 Operation as input signal high-/low-level width measurement.....	260
6.8.5 Operation as delay counter .....	264
<b>6.9 Simultaneous Channel Operation Function of Timer Array Unit .....</b>	<b>269</b>
6.9.1 Operation as one-shot pulse output function .....	269
6.9.2 Operation as PWM function.....	276
6.9.3 Operation as multiple PWM output function .....	283
6.9.4 Remote control output function.....	291
<b>6.10 Cautions When Using Timer Array Unit.....</b>	<b>294</b>
6.10.1 Cautions when using timer output .....	294
 <b>CHAPTER 7 16-BIT TIMER KB20.....</b>	 <b>295</b>
<b>7.1 Functions of 16-bit Timer KB20 .....</b>	<b>295</b>
<b>7.2 Configuration of 16-bit Timer KB20.....</b>	<b>296</b>
7.2.1 16-bit timer KB2 compare registers 00 to 03 (TKBCR00 to TKBCR03).....	298
<b>7.3 Registers Controlling 16-bit Timer KB20 .....</b>	<b>299</b>
7.3.1 Peripheral enable register (PER1).....	300
7.3.2 Timer clock select register (TKBPSCS0).....	301
7.3.3 16-bit timer KB2 operation control register 00 (TKBCTL00) .....	302
7.3.4 16-bit timer KB2 operation control register 01 (TKBCTL01) .....	304
7.3.5 16-bit timer KB2 output control register 00 (TKBIOC00) .....	305
7.3.6 16-bit timer KB2 output control register 01 (TKBIOC01) .....	306
7.3.7 16-bit timer KB2 flag register 0 (TKBFLG0).....	307
7.3.8 16-bit timer KB2 trigger register 0 (TKBTRG0).....	308
7.3.9 16-bit timer KB2 flag clear trigger register 0 (TKBCLR0).....	309
7.3.10 16-bit timer KB2 dithering count registers 00, 01 (TKBDNR00, TKBDNR01).....	310
7.3.11 16-bit timer KB2 compare 1L & dithering count register 00 (TKBCRLD00) .....	311
7.3.12 16-bit timer KB2 compare 3L & dithering count register 01 (TKBCRLD01) .....	311
7.3.13 16-bit timer KB2 smooth start initial duty registers 00, 01 (TKBSIR00, TKBSIR01) .....	312
7.3.14 16-bit timer KB2 smooth start step width registers 00, 01 (TKBSSR00, TKBSSR01) .....	312
7.3.15 16-bit timer KB2 maximum frequency limit setting register 0 (TKBMFR0).....	313
7.3.16 16-bit timer KB2 counter restart select register (ELSELRn) .....	314
7.3.17 16-bit timer KB2 output switch register (PWCTKB) .....	315
7.3.18 Forced output stop function control register 0 (TKBPACTL00).....	316

7.3.19	Forced output stop function control register 1 (TKBPACTL01).....	318
7.3.20	Forced output stop function control register 2 (TKBPACTL02).....	321
7.3.21	Forced output stop function flag register (TKBP AFLG0) .....	322
7.3.22	Forced output stop function 1 start register (TKBPAHFS0).....	323
7.3.23	Forced output stop function 1 stop register (TKBPAHFT0) .....	324
7.3.24	Registers controlling port functions of 16-bit timer KB2 output pins .....	325
<b>7.4</b>	<b>Operation of 16-bit Timer KB20 .....</b>	<b>326</b>
7.4.1	Counter basic operation .....	326
7.4.2	Default level and active level .....	327
7.4.3	Stop/restart operation.....	331
7.4.4	Batch overwrite operation.....	334
7.4.5	Standalone mode (period controlled by TKBCR00).....	335
7.4.6	Standalone mode (period controlled by external trigger input) .....	341
7.4.7	Interleave PFC (Power Factor Correction) output mode .....	348
<b>7.5</b>	<b>Option Functions of 16-bit Timer KB20.....</b>	<b>361</b>
7.5.1	PWM output dithering function .....	361
7.5.2	PWM output smooth start function .....	365
7.5.3	Maximum frequency limit function .....	368
7.5.4	PWM output function for IH control.....	369
<b>7.6</b>	<b>Forced Output Stop Function .....</b>	<b>372</b>
7.6.1	Forced output stop functions 1 and 2 .....	373
<b>7.7</b>	<b>Operation of Forced Output Stop Function 1 .....</b>	<b>375</b>
7.7.1	I/O setting for forced output stop function 1 .....	375
7.7.2	Basic operation of forced output stop function 1 .....	378
7.7.3	Notes on using forced output stop function 1 .....	382
<b>7.8</b>	<b>Operation of Forced Output Stop Function 2.....</b>	<b>383</b>
7.8.1	I/O setting for forced output stop function 2.....	383
7.8.2	Basic operation of forced output stop function 2.....	385
<b>7.9</b>	<b>Cautions When Using 16-bit Timer KB20.....</b>	<b>386</b>
7.9.1	Using 16-bit timer KB20 together with the LIN-bus functions .....	386
7.9.2	Cautions when using the counter restart trigger .....	386
7.9.3	Cautions when using the forced output stop function (when not using the PWM output function for IH control).....	386
7.9.4	Cautions when using the PWM output function for IH control.....	387
<b>CHAPTER 8</b>	<b>REAL-TIME CLOCK 2.....</b>	<b>388</b>
<b>8.1</b>	<b>Functions of Real-time Clock 2.....</b>	<b>388</b>
<b>8.2</b>	<b>Configuration of Real-time Clock 2 .....</b>	<b>388</b>
<b>8.3</b>	<b>Registers Controlling Real-time Clock 2.....</b>	<b>390</b>
8.3.1	Peripheral enable register 0 (PER0).....	391
8.3.2	Subsystem clock supply mode control register (OSMC) .....	392

8.3.3	Power-on-reset status register (PORSR) .....	393
8.3.4	Real-time clock control register 0 (RTCC0) .....	394
8.3.5	Real-time clock control register 1 (RTCC1) .....	396
8.3.6	Second count register (SEC) .....	399
8.3.7	Minute count register (MIN) .....	399
8.3.8	Hour count register (HOUR) .....	400
8.3.9	Date count register (DAY) .....	402
8.3.10	Day-of-week count register (WEEK) .....	403
8.3.11	Month count register (MONTH) .....	404
8.3.12	Year count register (YEAR) .....	404
8.3.13	Clock error correction register (SUBCUD) .....	405
8.3.14	Alarm minute register (ALARMWM) .....	408
8.3.15	Alarm hour register (ALARMWH) .....	408
8.3.16	Alarm day-of-week register (ALARMWW) .....	409
8.3.17	Registers controlling port functions of real-time clock 2 output pins .....	410
<b>8.4</b>	<b>Real-time Clock 2 Operation .....</b>	<b>411</b>
8.4.1	Starting operation of real-time clock 2 .....	411
8.4.2	Shifting to HALT/STOP mode after starting operation .....	412
8.4.3	Reading real-time clock 2 counter .....	413
8.4.4	Writing to real-time clock 2 counter .....	414
8.4.5	Setting alarm of real-time clock 2 .....	415
8.4.6	1 Hz output of real-time clock 2 .....	416
8.4.7	Clock error correction register setting procedure .....	417
8.4.8	Example of watch error correction of real-time clock 2 .....	418
<b>CHAPTER 9</b>	<b>12-BIT INTERVAL TIMER .....</b>	<b>420</b>
<b>9.1</b>	<b>Functions of 12-bit Interval Timer .....</b>	<b>420</b>
<b>9.2</b>	<b>Configuration of 12-bit Interval Timer .....</b>	<b>420</b>
<b>9.3</b>	<b>Registers Controlling 12-bit Interval Timer .....</b>	<b>421</b>
9.3.1	Peripheral enable register 1 (PER1) .....	421
9.3.2	Subsystem clock supply mode control register (OSMC) .....	422
9.3.3	12-bit interval timer control register (ITMC) .....	423
<b>9.4</b>	<b>12-bit Interval Timer Operation .....</b>	<b>424</b>
9.4.1	12-bit interval timer operation timing .....	424
9.4.2	Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode .....	425
<b>CHAPTER 10</b>	<b>CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER .....</b>	<b>426</b>
<b>10.1</b>	<b>Functions of Clock Output/Buzzer Output Controller .....</b>	<b>426</b>
<b>10.2</b>	<b>Configuration of Clock Output/Buzzer Output Controller .....</b>	<b>428</b>
<b>10.3</b>	<b>Registers Controlling Clock Output/Buzzer Output Controller .....</b>	<b>428</b>



10.3.1	Clock output select register n (CKSn) .....	428
10.3.2	Subsystem clock supply mode control register (OSMC).....	430
<b>10.4</b>	<b>Operations of Clock Output/Buzzer Output Controller .....</b>	<b>432</b>
10.4.1	Operation as output pin .....	432
<b>10.5</b>	<b>Cautions of Clock Output/Buzzer Output Controller .....</b>	<b>432</b>
<b>CHAPTER 11</b>	<b>WATCHDOG TIMER .....</b>	<b>433</b>
<b>11.1</b>	<b>Functions of Watchdog Timer.....</b>	<b>433</b>
<b>11.2</b>	<b>Configuration of Watchdog Timer .....</b>	<b>434</b>
<b>11.3</b>	<b>Register Controlling Watchdog Timer.....</b>	<b>435</b>
11.3.1	Watchdog timer enable register (WDTE).....	435
<b>11.4</b>	<b>Operation of Watchdog Timer .....</b>	<b>436</b>
11.4.1	Controlling operation of watchdog timer .....	436
11.4.2	Setting overflow time of watchdog timer .....	437
11.4.3	Setting window open period of watchdog timer .....	438
11.4.4	Setting watchdog timer interval interrupt .....	439
<b>CHAPTER 12</b>	<b>A/D CONVERTER .....</b>	<b>440</b>
<b>12.1</b>	<b>Function of A/D Converter.....</b>	<b>440</b>
<b>12.2</b>	<b>Configuration of A/D Converter .....</b>	<b>442</b>
<b>12.3</b>	<b>Registers Used in A/D Converter.....</b>	<b>444</b>
12.3.1	Peripheral enable register 0 (PER0).....	445
12.3.2	A/D converter mode register 0 (ADM0) .....	446
12.3.3	A/D converter mode register 1 (ADM1) .....	455
12.3.4	A/D converter mode register 2 (ADM2) .....	456
12.3.5	10-bit A/D conversion result register (ADCR) .....	458
12.3.6	8-bit A/D conversion result register (ADCRH) .....	459
12.3.7	Analog input channel specification register (ADS).....	460
12.3.8	Conversion result comparison upper limit setting register (ADUL) .....	461
12.3.9	Conversion result comparison lower limit setting register (ADLL) .....	461
12.3.10	A/D test register (ADTES) .....	462
12.3.11	Registers controlling port function of analog input pins .....	462
<b>12.4</b>	<b>A/D Converter Conversion Operations .....</b>	<b>463</b>
<b>12.5</b>	<b>Input Voltage and Conversion Results .....</b>	<b>465</b>
<b>12.6</b>	<b>A/D Converter Operation Modes.....</b>	<b>466</b>
12.6.1	Software trigger mode (sequential conversion mode) .....	466
12.6.2	Software trigger mode (one-shot conversion mode).....	467
12.6.3	Hardware trigger no-wait mode (sequential conversion mode) .....	468
12.6.4	Hardware trigger no-wait mode (one-shot conversion mode).....	469
12.6.5	Hardware trigger wait mode (sequential conversion mode).....	470
12.6.6	Hardware trigger wait mode (one-shot conversion mode).....	471

<b>12.7 A/D Converter Setup Flowchart .....</b>	<b>472</b>
12.7.1 Setting up software trigger mode.....	472
12.7.2 Setting up hardware trigger no-wait mode.....	473
12.7.3 Setting up hardware trigger wait mode.....	474
12.7.4 Setup when using temperature sensor (example for software trigger mode and one-shot conversion mode) .....	475
12.7.5 Setting up test mode .....	476
<b>12.8 SNOOZE Mode Function.....</b>	<b>477</b>
<b>12.9 How to Read A/D Converter Characteristics Table.....</b>	<b>480</b>
<b>12.10 Cautions for A/D Converter .....</b>	<b>482</b>
<b>CHAPTER 13 COMPARATOR.....</b>	<b>486</b>
<b>13.1 Functions of Comparator .....</b>	<b>486</b>
<b>13.2 Configuration of Comparator .....</b>	<b>487</b>
<b>13.3 Registers Controlling Comparator .....</b>	<b>488</b>
13.3.1 Peripheral enable register 1 (PER1).....	488
13.3.2 Comparator mode setting register (COMPMDR).....	489
13.3.3 Comparator filter control register (COMPFIR) .....	491
13.3.4 Comparator output control register (COMPOCR).....	492
13.3.5 Registers controlling port functions of analog input pins .....	493
<b>13.4 Operation.....</b>	<b>494</b>
13.4.1 Comparator i digital filter (i = 0 or 1) .....	496
13.4.2 Comparator i (i = 0 or 1) interrupts .....	496
13.4.3 Comparator i output (i = 0 or 1) .....	497
13.4.4 Stopping or supplying comparator clock.....	497
<b>CHAPTER 14 SERIAL ARRAY UNIT.....</b>	<b>498</b>
<b>14.1 Functions of Serial Array Unit.....</b>	<b>499</b>
14.1.1 3-wire serial I/O (CSI00, CSI10) .....	499
14.1.2 UART (UART0 to UART3).....	500
14.1.3 Simplified I <sup>2</sup> C (IIC00, IIC10).....	501
<b>14.2 Configuration of Serial Array Unit .....</b>	<b>502</b>
14.2.1 Shift register .....	505
14.2.2 Lower 8/9 bits of the serial data register mn (SDRmn).....	505
<b>14.3 Registers Controlling Serial Array Unit.....</b>	<b>507</b>
14.3.1 Peripheral enable register 0 (PER0).....	508
14.3.2 Serial clock select register m (SPSm) .....	509
14.3.3 Serial mode register mn (SMRmn) .....	510
14.3.4 Serial communication operation setting register mn (SCRmn) .....	511
14.3.5 Serial data register mn (SDRmn) .....	514
14.3.6 Serial flag clear trigger register mn (SIRmn) .....	516

14.3.7	Serial status register mn (SSRmn)	517
14.3.8	Serial channel start register m (SSm)	519
14.3.9	Serial channel stop register m (STm)	520
14.3.10	Serial channel enable status register m (SEm)	521
14.3.11	Serial output enable register m (SOEm)	522
14.3.12	Serial output register m (SOM)	523
14.3.13	Serial output level register m (SOLm)	524
14.3.14	Serial standby control register m (SSCm)	526
14.3.15	Input switch control register (ISC)	527
14.3.16	Noise filter enable register 0 (NFEN0)	528
14.3.17	Registers controlling port functions of serial I/O pins	529
<b>14.4</b>	<b>Operation Stop Mode</b>	<b>530</b>
14.4.1	Stopping the operation by units	530
14.4.2	Stopping the operation by channels	531
<b>14.5</b>	<b>Operation of 3-Wire Serial I/O (CSI00, CSI10) Communication</b>	<b>532</b>
14.5.1	Master transmission	534
14.5.2	Master reception	544
14.5.3	Master transmission/reception	553
14.5.4	Slave transmission	563
14.5.5	Slave reception	573
14.5.6	Slave transmission/reception	580
14.5.7	SNOOZE mode function	590
14.5.8	Calculating transfer clock frequency	594
14.5.9	Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI10) communication	596
<b>14.6</b>	<b>Operation of UART (UART0 to UART3) Communication</b>	<b>597</b>
14.6.1	UART transmission	599
14.6.2	UART reception	609
14.6.3	SNOOZE mode function	616
14.6.4	Calculating baud rate	624
14.6.5	Procedure for processing errors that occurred during UART (UART0 to UART3) communication	628
<b>14.7</b>	<b>LIN Communication Operation</b>	<b>629</b>
14.7.1	LIN transmission	629
14.7.2	LIN reception	632
<b>14.8</b>	<b>Operation of Simplified I<sup>2</sup>C (IIC00, IIC10) Communication</b>	<b>637</b>
14.8.1	Address field transmission	639
14.8.2	Data transmission	645
14.8.3	Data reception	649
14.8.4	Stop condition generation	654
14.8.5	Calculating transfer rate	655

14.8.6 Procedure for processing errors that occurred during simplified I <sup>2</sup> C (IIC00, IIC10) communication.....	657
<b>CHAPTER 15 SERIAL INTERFACE IICA .....</b>	<b>658</b>
<b>15.1 Functions of Serial Interface IICA.....</b>	<b>658</b>
<b>15.2 Configuration of Serial Interface IICA .....</b>	<b>661</b>
<b>15.3 Registers Controlling Serial Interface IICA.....</b>	<b>664</b>
15.3.1 Peripheral enable register 0 (PER0).....	664
15.3.2 IICA control register n0 (IICCTLn0) .....	665
15.3.3 IICA status register n (IICSn).....	670
15.3.4 IICA flag register n (IICFn).....	672
15.3.5 IICA control register n1 (IICCTLn1) .....	674
15.3.6 IICA low-level width setting register n (IICWLn) .....	676
15.3.7 IICA high-level width setting register n (IICWHn) .....	676
15.3.8 Port mode register 6 (PM6) .....	677
<b>15.4 I<sup>2</sup>C Bus Mode Functions.....</b>	<b>678</b>
15.4.1 Pin configuration.....	678
15.4.2 Setting transfer clock by using IICWLn and IICWHn registers.....	679
<b>15.5 I<sup>2</sup>C Bus Definitions and Control Methods .....</b>	<b>681</b>
15.5.1 Start conditions.....	681
15.5.2 Addresses .....	682
15.5.3 Transfer direction specification .....	682
15.5.4 Acknowledge (ACK) .....	683
15.5.5 Stop condition.....	684
15.5.6 Wait .....	685
15.5.7 Canceling wait .....	687
15.5.8 Interrupt request (INTIICAn) generation timing and wait control.....	688
15.5.9 Address match detection method .....	689
15.5.10 Error detection .....	689
15.5.11 Extension code .....	689
15.5.12 Arbitration .....	690
15.5.13 Wakeup function.....	692
15.5.14 Communication reservation.....	695
15.5.15 Cautions .....	699
15.5.16 Communication operations.....	700
15.5.17 Timing of I <sup>2</sup> C interrupt request (INTIICAn) occurrence .....	707
<b>15.6 Timing Charts .....</b>	<b>728</b>
<b>CHAPTER 16 LCD CONTROLLER/DRIVER .....</b>	<b>743</b>
<b>16.1 Functions of LCD Controller/Driver.....</b>	<b>744</b>
<b>16.2 Configuration of LCD Controller/Driver .....</b>	<b>746</b>

<b>16.3 Registers Controlling LCD Controller/Driver.....</b>	<b>748</b>
16.3.1 LCD mode register 0 (LCDM0).....	749
16.3.2 LCD mode register 1 (LCDM1).....	751
16.3.3 Subsystem clock supply mode control register (OSMC).....	753
16.3.4 LCD clock control register 0 (LCDC0).....	754
16.3.5 LCD boost level control register (VLCD).....	755
16.3.6 LCD input switch control register (ISCLCD).....	756
16.3.7 LCD port function registers 0 to 6 (PFSEG0 to PFSEG6).....	758
16.3.8 Port mode registers 0 to 5, 7, 13 (PM0 to PM5, PM7, PM13).....	763
<b>16.4 LCD Display Data Registers.....</b>	<b>764</b>
<b>16.5 Selection of LCD Display Register.....</b>	<b>768</b>
16.5.1 A-pattern area and B-pattern area data display.....	769
16.5.2 Blinking display (Alternately displaying A-pattern and B-pattern area data).....	769
<b>16.6 Setting the LCD Controller/Driver.....</b>	<b>770</b>
<b>16.7 Operation Stop Procedure.....</b>	<b>773</b>
<b>16.8 Supplying LCD Drive Voltages VL1, VL2, VL3, and VL4.....</b>	<b>774</b>
16.8.1 External resistance division method.....	774
16.8.2 Internal voltage boosting method.....	776
16.8.3 Capacitor split method.....	777
<b>16.9 Common and Segment Signals.....</b>	<b>778</b>
<b>16.10 Display Modes.....</b>	<b>786</b>
16.10.1 Static display example.....	786
16.10.2 Two-time-slice display example.....	789
16.10.3 Three-time-slice display example.....	792
16.10.4 Four-time-slice display example.....	796
16.10.5 Eight-time-slice display example.....	800
<b>CHAPTER 17 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR.....</b>	<b>804</b>
<b>17.1 Functions of Multiplier and Divider/Multiply-Accumulator.....</b>	<b>804</b>
<b>17.2 Configuration of Multiplier and Divider/Multiply-Accumulator.....</b>	<b>804</b>
17.2.1 Multiplication/division data register A (MDAH, MDAL).....	806
17.2.2 Multiplication/division data register B (MDBL, MDBH).....	807
17.2.3 Multiplication/division data register C (MDCL, MDCH).....	808
<b>17.3 Register Controlling Multiplier and Divider/Multiply-Accumulator.....</b>	<b>810</b>
17.3.1 Multiplication/division control register (MDUC).....	810
<b>17.4 Operations of Multiplier and Divider/Multiply-Accumulator.....</b>	<b>812</b>
17.4.1 Multiplication (unsigned) operation.....	812
17.4.2 Multiplication (signed) operation.....	813
17.4.3 Multiply-accumulation (unsigned) operation.....	814
17.4.4 Multiply-accumulation (signed) operation.....	816
17.4.5 Division operation.....	818

<b>CHAPTER 18 DMA CONTROLLER</b> .....	<b>820</b>
<b>18.1 Functions of DMA Controller</b> .....	<b>820</b>
<b>18.2 Configuration of DMA Controller</b> .....	<b>821</b>
18.2.1 DMA SFR address register n (DSAn) .....	821
18.2.2 DMA RAM address register n (DRAn) .....	822
18.2.3 DMA byte count register n (DBCn) .....	823
<b>18.3 Registers Controlling DMA Controller</b> .....	<b>824</b>
18.3.1 DMA mode control register n (DMCn) .....	825
18.3.2 DMA operation control register n (DRCn) .....	828
<b>18.4 Operation of DMA Controller</b> .....	<b>829</b>
18.4.1 Operation procedure .....	829
18.4.2 Transfer mode .....	830
18.4.3 Termination of DMA transfer .....	830
<b>18.5 Example of Setting of DMA Controller</b> .....	<b>831</b>
18.5.1 CSI consecutive transmission .....	831
18.5.2 Consecutive capturing of A/D conversion results .....	833
18.5.3 UART consecutive reception + ACK transmission.....	835
18.5.4 Holding DMA transfer pending by DWAITn bit .....	836
18.5.5 Forced termination by software .....	837
<b>18.6 Cautions on Using DMA Controller</b> .....	<b>839</b>
 <b>CHAPTER 19 INTERRUPT FUNCTIONS</b> .....	 <b>842</b>
<b>19.1 Interrupt Function Types</b> .....	<b>842</b>
<b>19.2 Interrupt Sources and Configuration</b> .....	<b>842</b>
<b>19.3 Registers Controlling Interrupt Functions</b> .....	<b>848</b>
19.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L) .....	851
19.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L).....	852
19.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L).....	854
19.3.4 External interrupt rising edge enable register (EGP0), external interrupt falling edge enable register (EGN0) .....	856
19.3.5 Program status word (PSW).....	857
<b>19.4 Interrupt Servicing Operations</b> .....	<b>858</b>
19.4.1 Maskable interrupt request acknowledgment .....	858
19.4.2 Software interrupt request acknowledgment .....	861
19.4.3 Multiple interrupt servicing.....	861
19.4.4 Interrupt request hold .....	865
 <b>CHAPTER 20 KEY INTERRUPT FUNCTION</b> .....	 <b>866</b>
<b>20.1 Functions of Key Interrupt</b> .....	<b>866</b>

<b>20.2 Configuration of Key Interrupt .....</b>	<b>867</b>
<b>20.3 Register Controlling Key Interrupt .....</b>	<b>868</b>
20.3.1 Key return control register (KRCTL) .....	868
20.3.2 Key return mode register (KRM0).....	869
20.3.3 Key return flag register (KRF).....	870
20.3.4 Port mode register 7 (PM7) .....	870
<b>CHAPTER 21 STANDBY FUNCTION .....</b>	<b>871</b>
<b>21.1 Standby Function .....</b>	<b>871</b>
<b>21.2 Registers Controlling Standby Function .....</b>	<b>872</b>
<b>21.3 Standby Function Operation .....</b>	<b>872</b>
21.3.1 HALT mode .....	872
21.3.2 STOP mode.....	878
21.3.3 SNOOZE mode .....	884
<b>CHAPTER 22 RESET FUNCTION.....</b>	<b>887</b>
<b>22.1 Timing of Reset Operation .....</b>	<b>889</b>
<b>22.2 States of Operation During Reset Periods.....</b>	<b>891</b>
<b>22.3 Register for Confirming Reset Source .....</b>	<b>893</b>
22.3.1 Reset control flag register (RESF).....	893
22.3.2 Power-on-reset status register (PORSR) .....	894
<b>CHAPTER 23 POWER-ON-RESET CIRCUIT .....</b>	<b>896</b>
<b>23.1 Functions of Power-on-reset Circuit .....</b>	<b>896</b>
<b>23.2 Configuration of Power-on-reset Circuit.....</b>	<b>897</b>
<b>23.3 Operation of Power-on-reset Circuit .....</b>	<b>897</b>
<b>CHAPTER 24 VOLTAGE DETECTOR.....</b>	<b>901</b>
<b>24.1 Functions of Voltage Detector .....</b>	<b>901</b>
<b>24.2 Configuration of Voltage Detector.....</b>	<b>902</b>
<b>24.3 Registers Controlling Voltage Detector .....</b>	<b>902</b>
24.3.1 Voltage detection register (LVIM) .....	903
24.3.2 Voltage detection level register (LVIS) .....	904
<b>24.4 Operation of Voltage Detector .....</b>	<b>907</b>
24.4.1 When used as reset mode.....	907
24.4.2 When used as interrupt mode .....	909
24.4.3 When used as interrupt and reset mode .....	911
<b>24.5 Cautions for Voltage Detector.....</b>	<b>916</b>
<b>CHAPTER 25 SAFETY FUNCTIONS.....</b>	<b>918</b>

<b>25.1 Overview of Safety Functions .....</b>	<b>918</b>
<b>25.2 Registers Used by Safety Functions .....</b>	<b>919</b>
<b>25.3 Operation of Safety Functions .....</b>	<b>919</b>
25.3.1 Flash memory CRC operation function (high-speed CRC).....	919
25.3.1.1 Flash memory CRC control register (CRC0CTL).....	920
25.3.1.2 Flash memory CRC operation result register (PGCRCL).....	921
25.3.2 CRC operation function (general-purpose CRC).....	923
25.3.2.1 CRC input register (CRCIN).....	923
25.3.2.2 CRC data register (CRCD) .....	924
25.3.3 RAM parity error detection function .....	925
25.3.3.1 RAM parity error control register (RPECTL) .....	925
25.3.4 RAM guard function.....	926
25.3.4.1 Invalid memory access detection control register (IAWCTL) .....	926
25.3.5 SFR guard function .....	927
25.3.5.1 Invalid memory access detection control register (IAWCTL) .....	927
25.3.6 Invalid memory access detection function.....	928
25.3.6.1 Invalid memory access detection control register (IAWCTL) .....	929
25.3.7 Frequency detection function .....	930
25.3.7.1 Timer input select register 0 (TIS0).....	931
25.3.8 A/D test function .....	932
25.3.8.1 A/D test register (ADTES).....	933
25.3.8.2 Analog input channel specification register (ADS) .....	933
25.3.9 Digital output signal level detection function for I/O pins .....	935
25.3.9.1 Port mode select register (PMS).....	935
 <b>CHAPTER 26 REGULATOR .....</b>	 <b>936</b>
<b>26.1 Regulator Overview.....</b>	<b>936</b>
 <b>CHAPTER 27 OPTION BYTE.....</b>	 <b>937</b>
<b>27.1 Functions of Option Bytes .....</b>	<b>937</b>
27.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H).....	937
27.1.2 On-chip debug option byte (000C3H/ 010C3H).....	938
<b>27.2 Format of User Option Byte .....</b>	<b>939</b>
<b>27.3 Format of On-chip Debug Option Byte.....</b>	<b>943</b>
<b>27.4 Setting of Option Byte.....</b>	<b>944</b>
 <b>CHAPTER 28 FLASH MEMORY .....</b>	 <b>945</b>
<b>28.1 Serial Programming Using Flash Memory Programmer .....</b>	<b>947</b>
28.1.1 Programming environment .....	949
28.1.2 Communication mode .....	949



<b>28.2 Serial Programming Using External Device (that Incorporates UART)</b> .....	<b>950</b>
28.2.1 Programming environment .....	950
28.2.2 Communication mode .....	951
<b>28.3 Connection of Pins on Board</b> .....	<b>952</b>
28.3.1 P40/TOOL0 pin .....	952
28.3.2 RESET pin.....	952
28.3.3 Port pins .....	953
28.3.4 REGC pin .....	953
28.3.5 X1 and X2 pins .....	953
28.3.6 Power supply .....	953
<b>28.4 Serial Programming Method</b> .....	<b>954</b>
28.4.1 Controlling serial programming.....	954
28.4.2 Flash memory programming mode.....	955
28.4.3 Selecting communication mode.....	956
28.4.4 Communication commands .....	957
<b>28.5 Self-Programming</b> .....	<b>959</b>
28.5.1 Self-programming procedure.....	960
28.5.2 Boot swap function .....	961
28.5.3 Flash shield window function.....	963
<b>28.6 Security Settings</b> .....	<b>964</b>
<b>28.7 Data Flash</b> .....	<b>966</b>
28.7.1 Data flash overview .....	966
28.7.2 Register controlling data flash memory .....	967
28.7.2.1 Data flash control register (DFLCTL) .....	967
28.7.3 Procedure for accessing data flash memory .....	968
<b>CHAPTER 29 ON-CHIP DEBUG FUNCTION</b> .....	<b>969</b>
<b>29.1 Connecting E1 On-chip Debugging Emulator</b> .....	<b>969</b>
<b>29.2 On-Chip Debug Security ID</b> .....	<b>970</b>
<b>29.3 Securing of User Resources</b> .....	<b>970</b>
<b>CHAPTER 30 BCD CORRECTION CIRCUIT</b> .....	<b>972</b>
<b>30.1 BCD Correction Circuit Function</b> .....	<b>972</b>
<b>30.2 Registers Used by BCD Correction Circuit</b> .....	<b>972</b>
30.2.1 BCD correction result register (BCDADJ).....	972
<b>30.3 BCD Correction Circuit Operation</b> .....	<b>973</b>
<b>CHAPTER 31 INSTRUCTION SET</b> .....	<b>975</b>
<b>31.1 Conventions Used in Operation List</b> .....	<b>976</b>
31.1.1 Operand identifiers and specification methods.....	976

31.1.2	Description of operation column .....	977
31.1.3	Description of flag operation column .....	978
31.1.4	PREFIX instruction .....	978
<b>31.2</b>	<b>Operation List .....</b>	<b>979</b>
<b>CHAPTER 32</b>	<b>ELECTRICAL SPECIFICATIONS (TA = -40 to +85°C) .....</b>	<b>996</b>
<b>32.1</b>	<b>Absolute Maximum Ratings .....</b>	<b>997</b>
<b>32.2</b>	<b>Oscillator Characteristics .....</b>	<b>1000</b>
32.2.1	X1 and XT1 oscillator characteristics .....	1000
32.2.2	On-chip oscillator characteristics .....	1001
<b>32.3</b>	<b>DC Characteristics .....</b>	<b>1002</b>
32.3.1	Pin characteristics .....	1002
32.3.2	Supply current characteristics .....	1007
<b>32.4</b>	<b>AC Characteristics .....</b>	<b>1013</b>
<b>32.5</b>	<b>Peripheral Functions Characteristics .....</b>	<b>1018</b>
32.5.1	Serial array unit .....	1018
32.5.2	Serial interface IICA .....	1040
<b>32.6</b>	<b>Analog Characteristics .....</b>	<b>1045</b>
32.6.1	A/D converter characteristics .....	1045
32.6.2	Temperature sensor /internal reference voltage characteristics .....	1049
32.6.3	Comparator characteristics .....	1049
32.6.4	POR circuit characteristics .....	1050
32.6.5	LVD circuit characteristics .....	1051
32.6.6	Supply voltage rising slope characteristics .....	1052
<b>32.7</b>	<b>LCD Characteristics .....</b>	<b>1053</b>
32.7.1	External resistance division method .....	1053
32.7.2	Internal voltage boosting method .....	1054
32.7.3	Capacitor split method .....	1056
<b>32.8</b>	<b>RAM Data Retention Characteristics .....</b>	<b>1057</b>
<b>32.9</b>	<b>Flash Memory Programming Characteristics .....</b>	<b>1057</b>
<b>32.10</b>	<b>Dedicated Flash Memory Programmer Communication (UART) .....</b>	<b>1057</b>
<b>32.11</b>	<b>Timing Specifications for Switching Flash Memory Programming Modes .....</b>	<b>1058</b>
<b>CHAPTER 33</b>	<b>ELECTRICAL SPECIFICATIONS</b>	
	<b>(G: INDUSTRIAL APPLICATIONS TA = -40 to +105°C) .....</b>	<b>1059</b>
<b>33.1</b>	<b>Absolute Maximum Ratings .....</b>	<b>1061</b>
<b>33.2</b>	<b>Oscillator Characteristics .....</b>	<b>1064</b>
33.2.1	X1 and XT1 oscillator characteristics .....	1064
33.2.2	On-chip oscillator characteristics .....	1065
<b>33.3</b>	<b>DC Characteristics .....</b>	<b>1066</b>
33.3.1	Pin characteristics .....	1066

33.3.2 Supply current characteristics .....	1071
<b>33.4 AC Characteristics .....</b>	<b>1077</b>
<b>33.5 Peripheral Functions Characteristics.....</b>	<b>1080</b>
33.5.1 Serial array unit .....	1080
33.5.2 Serial interface IICA .....	1099
<b>33.6 Analog Characteristics .....</b>	<b>1100</b>
33.6.1 A/D converter characteristics.....	1100
33.6.2 Temperature sensor/internal reference voltage characteristics .....	1104
33.6.3 Comparator .....	1104
33.6.4 POR circuit characteristics .....	1105
33.6.5 LVD circuit characteristics .....	1106
33.6.6 Supply voltage rise time .....	1107
<b>33.7 LCD Characteristics .....</b>	<b>1108</b>
33.7.1 External resistance division method.....	1108
33.7.2 Internal voltage boosting method .....	1109
33.7.3 Capacitor split method.....	1111
<b>33.8 RAM Data Retention Characteristics.....</b>	<b>1112</b>
<b>33.9 Flash Memory Programming Characteristics.....</b>	<b>1112</b>
<b>33.10 Dedicated Flash Memory Programmer Communication (UART).....</b>	<b>1112</b>
<b>33.11 Timing Specifications for Switching Flash Memory Programming Modes.....</b>	<b>1113</b>
<b>CHAPTER 34 PACKAGE DRAWINGS .....</b>	<b>1114</b>
34.1 64-pin Products .....	1114
34.2 80-pin Products .....	1116
<b>APPENDIX A REVISION HISTORY .....</b>	<b>1118</b>
A.1 Major Revisions in This Edition .....	1118
A.2 Revision History of Preceding Editions .....	1121

## CHAPTER 1 OUTLINE

### 1.1 Features

#### Ultra-low power consumption technology

- $V_{DD}$  = single power supply voltage of 1.6 to 5.5 V which can operate a 1.8 V device at a low voltage
- 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  $\mu$ s: @ 24 MHz operation with high-speed on-chip oscillator) to ultra-low speed (30.5  $\mu$ s: @ 32.768 kHz operation with subsystem clock)
- Address space: 1 MB
- General-purpose registers: (8-bit register  $\times$  8)  $\times$  4 banks
- On-chip RAM: 1 to 8 KB

#### Code flash memory

- Code flash memory: 16 to 128 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: 4 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:  $V_{DD}$  = 1.8 to 5.5 V

#### High-speed on-chip oscillator

- Select from 48 MHz, 24 MHz, 16 MHz, 12 MHz, 8 MHz, 6 MHz, 4 MHz, 3 MHz, 2 MHz, and 1 MHz
- High accuracy:  $\pm 1.0\%$  ( $V_{DD}$  = 1.8 to 5.5 V,  $T_A$  =  $-20$  to  $+85^\circ\text{C}$ )

#### Operating ambient temperature

- $T_A$  =  $-40$  to  $+85^\circ\text{C}$  (A: Consumer applications)
- $T_A$  =  $-40$  to  $+105^\circ\text{C}$  (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 14 levels)

## DMA (Direct Memory Access) controller

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

## Multiplier and divider/multiply-accumulator

- 16 bits  $\times$  16 bits = 32 bits (Unsigned or signed)
- 32 bits  $\div$  32 bits = 32 bits (Unsigned)
- 16 bits  $\times$  16 bits + 32 bits = 32 bits (Unsigned or signed)

## Serial interface

- CSI: 2 channels
- UART/UART (LIN-bus supported): 3, 4 channels/1 channel
- I<sup>2</sup>C/Simplified I<sup>2</sup>C communication: 1 channel/2 channels

## Timer

- |     |                           |  |
|-----|---------------------------|--|
| <R> | • 16-bit timer:           | 8 channels (with remote control output function)                                 |
| <R> | • 16-bit timer KB20 (IH): | 1 channel (IH-only PWM output function)  |
|     | • 12-bit interval timer:  | 1 channel  |
|     | • Real-time clock 2:      | 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 ( $V_{DD} = 1.6$  to  $5.5$  V)
- Analog input: 9 to 12 channels
- Internal reference voltage (1.45 V) and temperature sensor<sup>Note 1</sup>

## Comparator

- 2 channels
- Operation mode: Comparator high-speed mode, comparator low-speed mode, or window mode
- External reference voltage and internal reference voltage are selectable

## LCD controller/driver

- Segment signal output: 36 (32)<sup>Note 2</sup> to 51 (47)<sup>Note 2</sup>
- Common signal output: 4 (8)<sup>Note 2</sup>
- Internal voltage boosting method, capacitor split method, and external resistance division method are switchable

## I/O port

- I/O port: 49 to 65 (N-ch open drain I/O [withstand voltage of 6 V]: 2, N-ch open drain I/O [ $V_{DD}$  withstand voltage]: 12 to 18)
- Can be set to N-ch open drain, TTL input buffer, and on-chip pull-up resistor
- Different potential interface: Can connect to a 1.8/2.5/3 V device
- On-chip key interrupt function
- On-chip clock output/buzzer output controller

## Others

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

- Notes**
1. Can be selected only in HS (high-speed main) mode
  2. The values in parentheses are the number of signal outputs when 8 com is used.

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

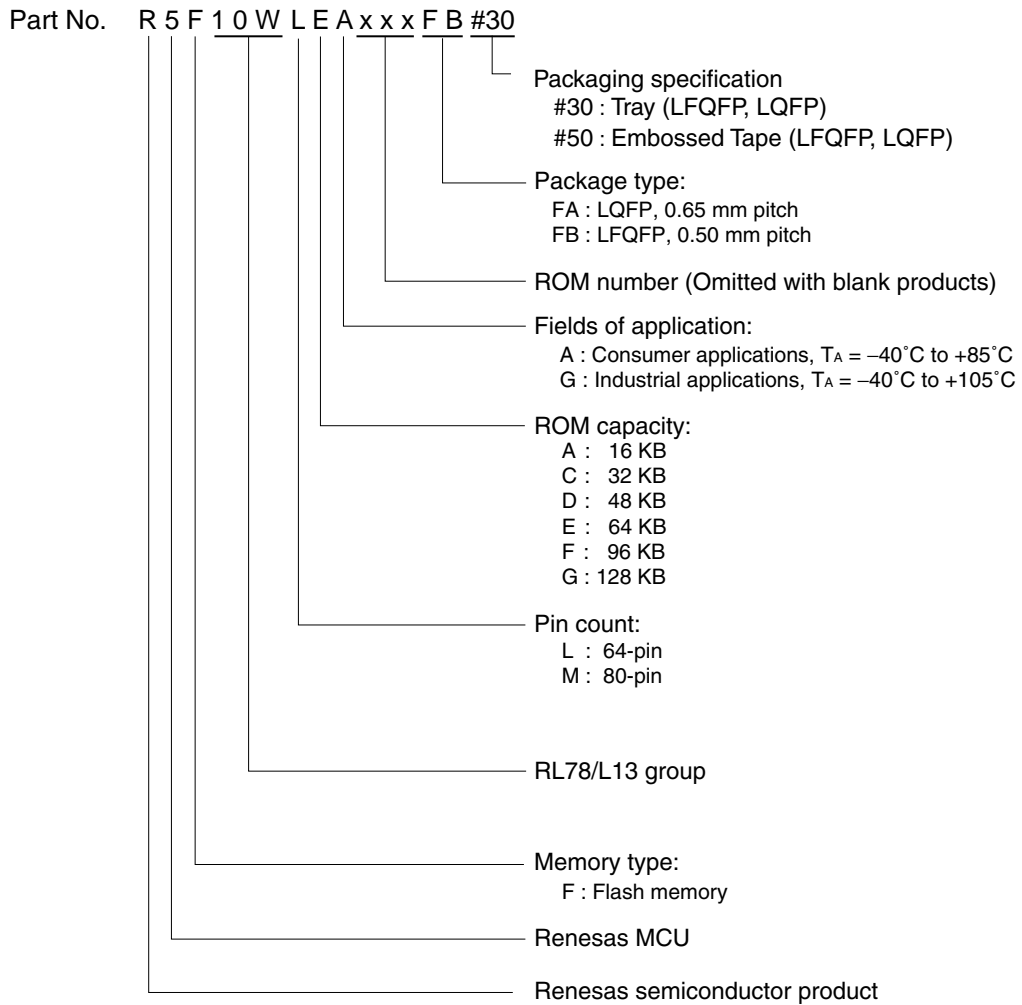
## ○ ROM, RAM capacities

Flash ROM	Data Flash	RAM	RL78/L13	
			64 pins	80 pins
128 KB	4 KB	8 KB <sup>Note</sup>	R5F10WLG	R5F10WMG
96 KB	4 KB	6 KB	R5F10WLF	R5F10WMF
64 KB	4 KB	4 KB	R5F10WLE	R5F10WME
48 KB	4 KB	2 KB	R5F10WLD	R5F10WMD
32 KB	4 KB	1.5 KB	R5F10WLC	R5F10WMC
16 KB	4 KB	1 KB	R5F10WLA	R5F10WMA

**Note** This is about 7 KB when the self-programming function and data flash function are used. (For details, see **CHAPTER 3.**)

1.2 List of Part Numbers

Figure 1-1. Part Number, Memory Size, and Package of RL78/L13



Pin Count	Package	Data Flash	Fields of Application <sup>Note</sup>	Ordering Part Number
64 pins	64-pin plastic LQFP (12 × 12 mm, 0.65 mm pitch)	Mounted	A	R5F10WLAFA#30, R5F10WLAFA#50, R5F10WLCAFA#30, R5F10WLCAFA#50, R5F10WLDAFA#30, R5F10WLDAFA#50, R5F10WLEAFA#30, R5F10WLEAFA#50, R5F10WLFafa#30, R5F10WLFafa#50, R5F10WLGafa#30, R5F10WLGafa#50
	64-pin plastic LFQFP (10 × 10 mm, 0.5 mm pitch)	Mounted	A  G	R5F10WLAafb#30, R5F10WLAafb#50, R5F10WLCAfb#30, R5F10WLCAfb#50, R5F10WLDAfb#30, R5F10WLDAfb#50, R5F10WLEafb#30, R5F10WLEafb#50, R5F10WLFafb#30, R5F10WLFafb#50, R5F10WLGafb#30, R5F10WLGafb#50, R5F10WLAGfb#30, R5F10WLAGfb#50, R5F10WLCGfb#30, R5F10WLCGfb#50, R5F10WLDGfb#30, R5F10WLDGfb#50, R5F10WLEGfb#30, R5F10WLEGfb#50, R5F10WLFgfb#30, R5F10WLFgfb#50, R5F10WLGgfb#30, R5F10WLGgfb#50
80 pins	80-pin plastic LQFP (14 × 14 mm, 0.65 mm pitch)	Mounted	A	R5F10WMAAFA#30, R5F10WMAAFA#50, R5F10WMCAFA#30, R5F10WMCAFA#50, R5F10WMDAFA#30, R5F10WMDAFA#50, R5F10WMEAFA#30, R5F10WMEAFA#50, R5F10WMFAFA#30, R5F10WMFAFA#50, R5F10WMGAFA#30, R5F10WMGAFA#50
	80-pin plastic LFQFP (12 × 12 mm, 0.5 mm pitch)	Mounted	A  G	R5F10WMAafb#30, R5F10WMAafb#50, R5F10WMCAfb#30, R5F10WMCAfb#50, R5F10WMDafb#30, R5F10WMDafb#50, R5F10WMEAfb#30, R5F10WMEAfb#50, R5F10WMFAfb#30, R5F10WMFAfb#50, R5F10WMGAfb#30, R5F10WMGAfb#50, R5F10WMAGfb#30, R5F10WMAGfb#50, R5F10WMCgfb#30, R5F10WMCgfb#50, R5F10WMDgfb#30, R5F10WMDgfb#50, R5F10WMEgfb#30, R5F10WMEgfb#50, R5F10WMFGfb#30, R5F10WMFGfb#50, R5F10WMGGfb#30, R5F10WMGGfb#50

**Note** For the fields of application, see **Figure 1-1 Part Number, Memory Size, and Package of RL78/L13**.

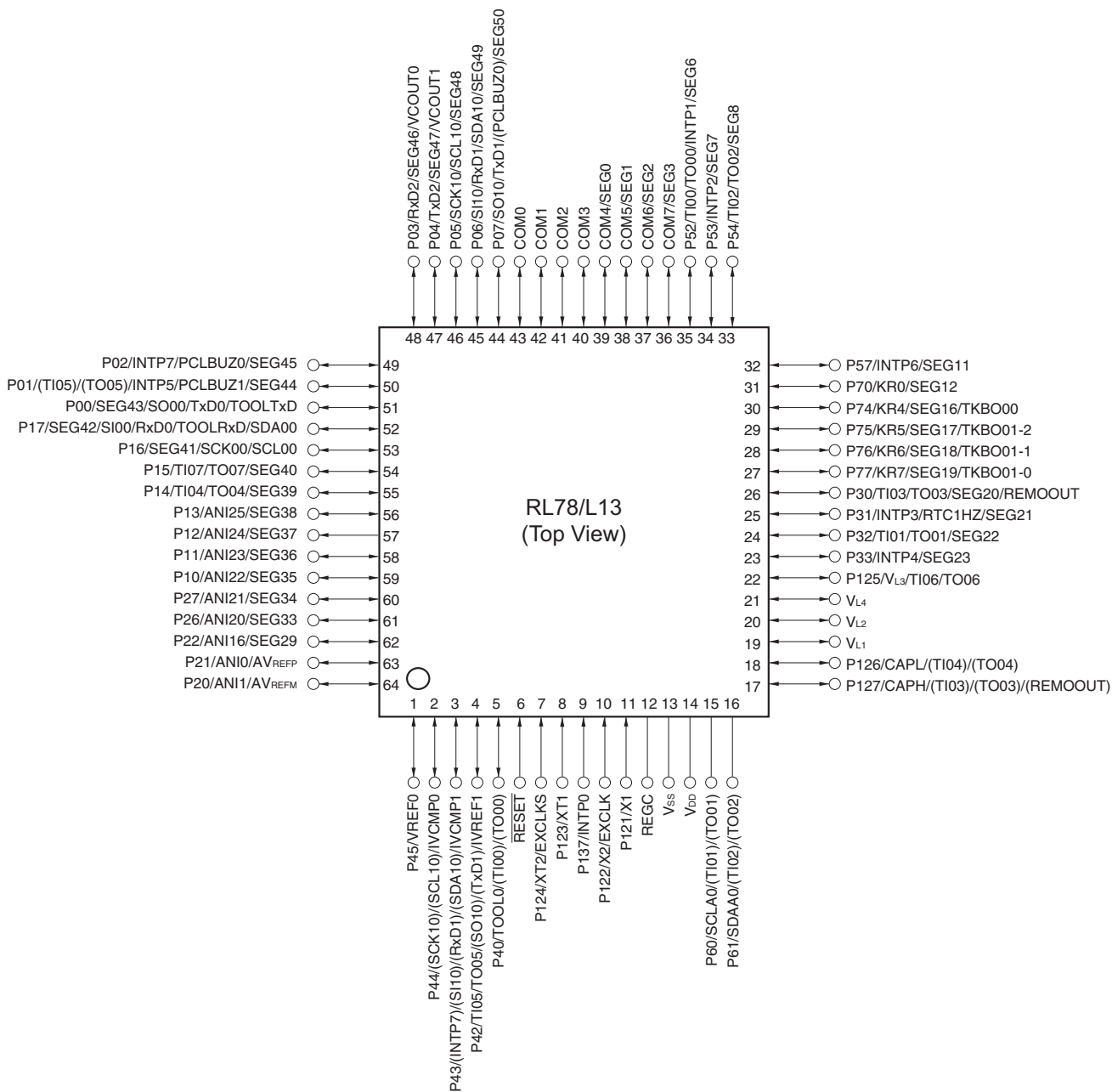
**Caution** The ordering part numbers represent the numbers at the time of publication. For the latest ordering part numbers, refer to the target product page of the Renesas Electronics website.



### 1.3 Pin Configuration (Top View)

#### <R> 1.3.1 64-pin products

- 64-pin plastic LQFP (12 × 12 mm, 0.65 mm pitch)
- 64-pin plastic LFQFP (10 × 10 mm, 0.5 mm pitch)

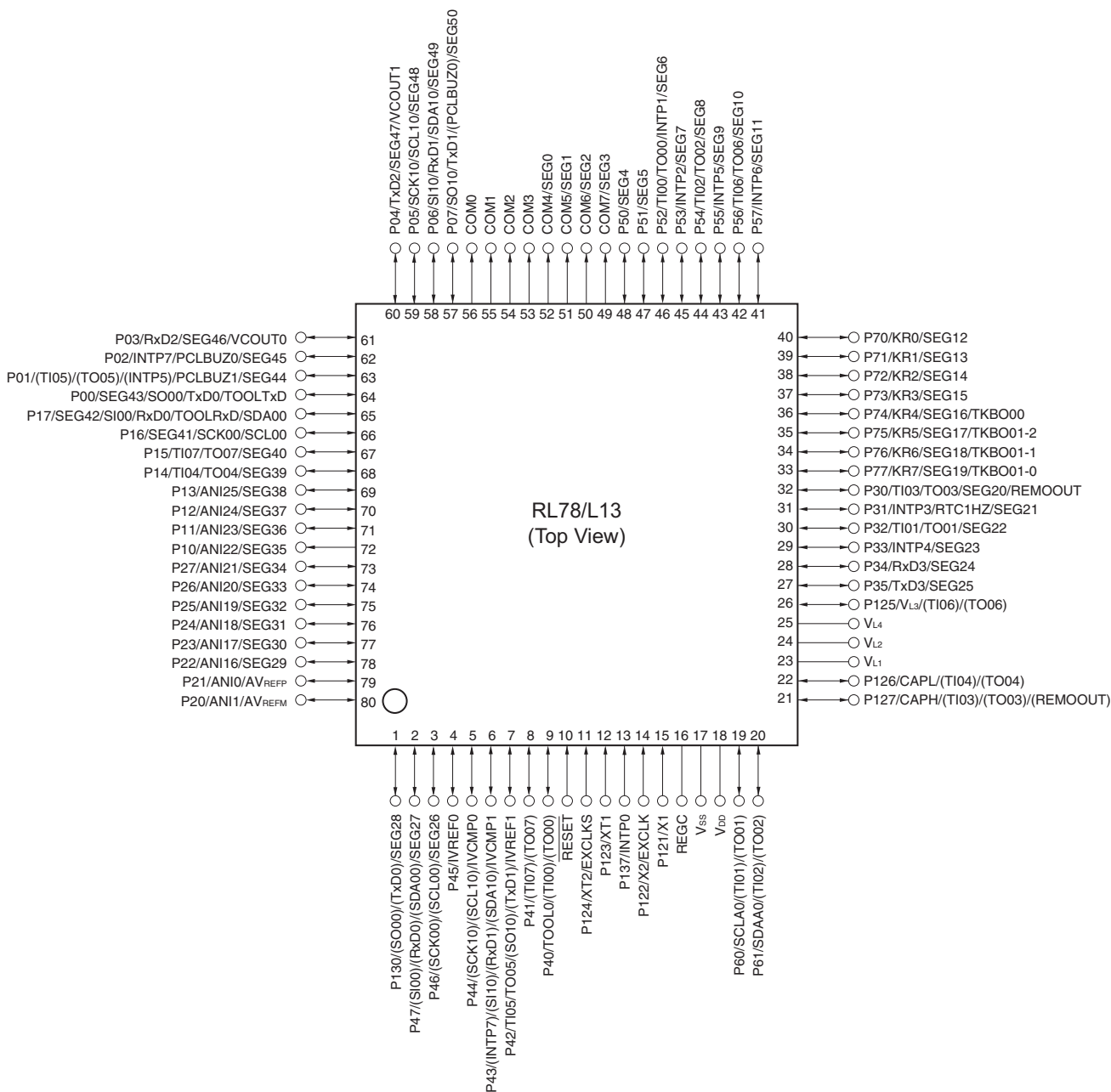


**Caution** Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

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

<R> 1.3.2 80-pin products

- 80-pin plastic LQFP (14 × 14 mm, 0.65 mm pitch)
- 80-pin plastic LFQFP (12 × 12 mm, 0.5 mm pitch)



**Caution** Connect the REGC pin to V<sub>SS</sub> via a capacitor (0.47 to 1 μF).

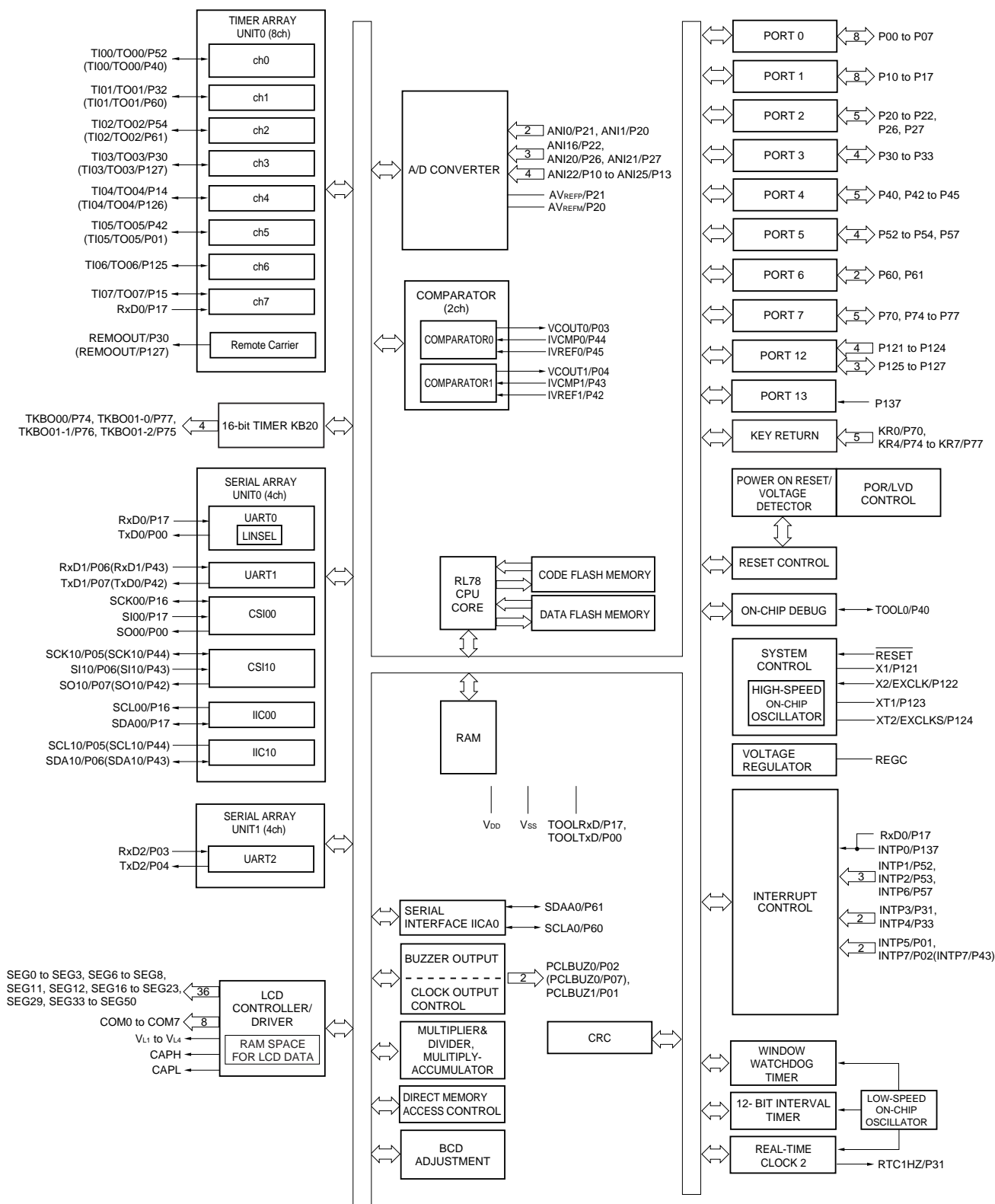
- Remarks**
1. For pin identification, see 1.4 Pin Identification.
  2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). See Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR).

## 1.4 Pin Identification

ANI0, ANI1,		PCLBUZ0, PCLBUZ1:	Programmable Clock Output/ Buzzer Output
ANI16 to ANI25:	Analog Input	REGC:	Regulator Capacitance
AVREFM:	Analog Reference Voltage Minus	REMOOUT:	Remote control Output
AVREFP:	Analog Reference Voltage Plus	RESET:	Reset
CAPH, CAPL:	Capacitor for LCD	RTC1HZ:	Real-time Clock 2 Correction Clock (1 Hz) Output
COM0 to COM7:	LCD Common Output	RxD0 to RxD3:	Receive Data
EXCLK:	External Clock Input (Main System Clock)	SCK00, SCK10, SCLA0:	Serial Clock Input/Output
EXCLKS:	External Clock Input (Subsystem Clock)	SCL00, SCL10:	Serial Clock Output
INTP0 to INTP7:	External Interrupt Input	SDAA0, SDA00, SDA10:	Serial Data Input/Output
IVCMP0, IVCMP1:	Comparator Input	SEG0 to SEG50:	LCD Segment Output
IVREF0, IVREF1:	Comparator Reference Input	SI00, SI10:	Serial Data Input
KR0 to KR7:	Key Return	SO00, SO10:	Serial Data Output
P00 to P07:	Port 0	TI00 to TI07:	Timer Input
P10 to P17:	Port 1	TO00 to TO07,	
P20 to P27:	Port 2	TKBO00, TKBO01-0,	
P30 to P35:	Port 3	TKBO01-1, TKBO01-2:	Timer Output
P40 to P47:	Port 4	TOOL0:	Data Input/Output for Tool
P50 to P57:	Port 5	TOOLRxD, TOOLTxD:	Data Input/Output for External Device
P60, P61:	Port 6	TxD0 to TxD3:	Transmit Data
P70 to P77:	Port 7	VCOU0, VCOU1:	Comparator Output
P121 to P127:	Port 12	VDD:	Power Supply
P130, P137:	Port 13	VL1 to VL4:	LCD Power Supply
		VSS:	Ground
		X1, X2:	Crystal Oscillator (Main System Clock)
		XT1, XT2:	Crystal Oscillator (Subsystem Clock)

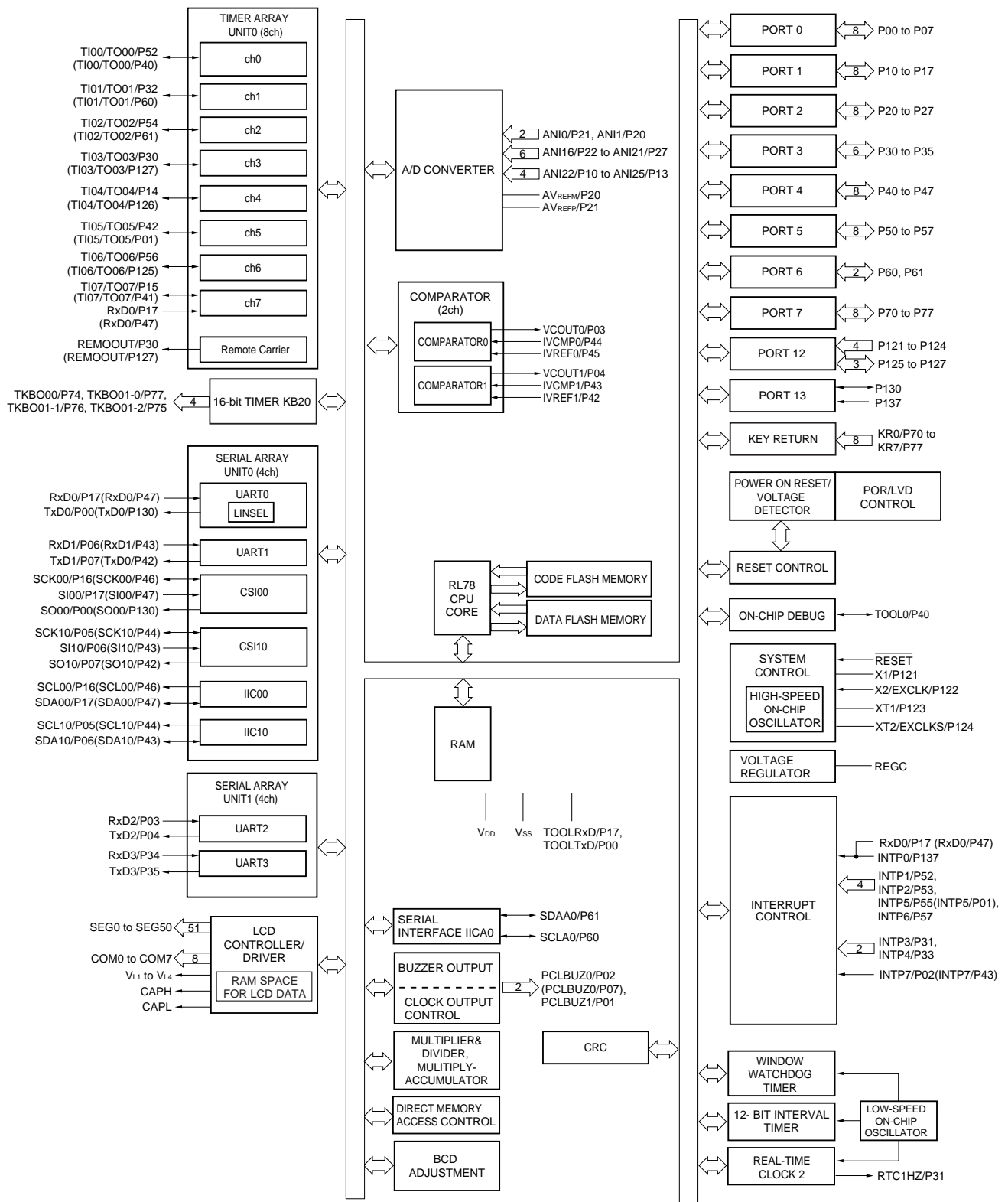
### 1.5 Block Diagram

#### 1.5.1 64-pin products



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

1.5.2 80-pin products



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

## 1.6 Outline of Functions

(1/2)

Item		64-pin	80-pin
		R5F10WLx (x = A, C-G)	R5F10WMx (x = A, C-G)
Code flash memory (KB)		16 to 128	16 to 128
Data flash memory (KB)		4	4
RAM (KB)		1 to 8 <sup>Note 1</sup>	1 to 8 <sup>Note 1</sup>
Address space		1 MB	
<R>	Main 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), LS (Low-speed main) mode: 1 to 8 MHz ( $V_{DD} = 1.8$ to $5.5$ V), LV (Low-voltage main) mode: 1 to 4 MHz ( $V_{DD} = 1.6$ to $5.5$ V)	
	High-speed on-chip oscillator	HS (High-speed main) mode: 1 to 24 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), LS (Low-speed main) mode: 1 to 8 MHz ( $V_{DD} = 1.8$ to $5.5$ V), LV (Low-voltage main) mode: 1 to 4 MHz ( $V_{DD} = 1.6$ to $5.5$ V)	
Clock for 16-bit timer KB20		48 MHz (TYP.): $V_{DD} = 2.7$ to $5.5$ V	
Subsystem clock		XT1 (crystal) oscillation, external subsystem clock input (EXCLKS) 32.768 kHz (TYP.): $V_{DD} = 1.6$ to $5.5$ V	
Low-speed on-chip oscillator		15 kHz (TYP.)	
General-purpose register		(8-bit register $\times$ 8) $\times$ 4 banks	
Minimum instruction execution time		0.04167 $\mu$ s (High-speed on-chip oscillator: $f_{IH} = 24$ MHz operation)	
		0.05 $\mu$ s (High-speed system clock: $f_{MX} = 20$ MHz operation)	
		30.5 $\mu$ s (Subsystem clock: $f_{SUB} = 32.768$ kHz operation)	
Instruction set		<ul style="list-style-type: none"> <li>• Data transfer (8/16 bits)</li> <li>• Adder and subtractor/logical operation (8/16 bits)</li> <li>• Multiplication (8 bits <math>\times</math> 8 bits)</li> <li>• Rotate, barrel shift, and bit manipulation (Set, reset, test, and Boolean operation), etc.</li> </ul>	
I/O port	Total	49	65
	CMOS I/O	42 (N-ch O.D. I/O [ $V_{DD}$ withstand voltage]: 12)	58 (N-ch O.D. I/O [ $V_{DD}$ withstand voltage]: 18)
	CMOS input	5	5
	CMOS output	–	–
	N-ch O.D I/O (withstand voltage: 6 V)	2	2
Timer	16-bit timer TAU	8 channels	
	16-bit timer KB20	1 channel	
	Watchdog timer	1 channel	
	12-bit interval timer (IT)	1 channel	
	Real-time clock 2	1 channel	
	RTC2 output	1 • 1 Hz (subsystem clock: $f_{SUB} = 32.768$ kHz)	
	Timer output	8 channels (PWM outputs: 7 <sup>Note 2</sup> ) (TAU used) 1 channel (timer KB20 used)	
	Remote control output function	1 (TAU used)	

- Notes**
1. In the case of the 8 KB, this is about 7 KB when the self-programming function and data flash function are used.
  2. The number of outputs varies depending on the setting of the channels in use and the number of master channels (see 6.9.3 Operation as multiple PWM output function).

(2/2)

Item	64-pin		80-pin	
	R5F10WLx (x = A, C-G)		R5F10WMx (x = A, C-G)	
Clock output/buzzer output controller	2			
	<ul style="list-style-type: none"> <li>2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (Main system clock: <math>f_{MAIN} = 20</math> MHz operation)</li> <li>256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (Subsystem clock: <math>f_{SUB} = 32.768</math> kHz operation)</li> </ul>			
8/10-bit resolution A/D converter	9 channels		12 channels	
Comparator	2 channels			
Serial interface	[64-pin] <ul style="list-style-type: none"> <li>CSI: 1 channel/UART (UART supporting LIN-bus): 1 channel/simplified I<sup>2</sup>C: 1 channel</li> <li>CSI: 1 channel/UART: 1 channel/simplified I<sup>2</sup>C: 1 channel</li> <li>UART: 1 channel</li> </ul> [80-pin] <ul style="list-style-type: none"> <li>CSI: 1 channel/UART (UART supporting LIN-bus): 1 channel/simplified I<sup>2</sup>C: 1 channel</li> <li>CSI: 1 channel/UART: 1 channel/simplified I<sup>2</sup>C: 1 channel</li> <li>UART: 2 channels</li> </ul>			
I <sup>2</sup> C bus	1 channel			
LCD controller/driver	Internal voltage boosting method, capacitor split method, and external resistance division method are switchable.			
Segment signal output	36 (32) <sup>Note 1</sup>		51 (47) <sup>Note 1</sup>	
Common signal output	4 (8) <sup>Note 1</sup>			
Multiplier and divider/multiply-accumulator	<ul style="list-style-type: none"> <li>16 bits × 16 bits = 32 bits (Unsigned or signed)</li> <li>32 bits ÷ 32 bits = 32 bits (Unsigned)</li> <li>16 bits × 16 bits + 32 bits = 32 bits (Unsigned or signed)</li> </ul>			
DMA controller	4 channels			
Vectored interrupt sources	Internal	32		35
	External	11		11
Key interrupt	5		8	
Reset	<ul style="list-style-type: none"> <li>Reset by RESET pin</li> <li>Internal reset by watchdog timer</li> <li>Internal reset by power-on-reset</li> <li>Internal reset by voltage detector</li> <li>Internal reset by illegal instruction execution<sup>Note 2</sup></li> <li>Internal reset by RAM parity error</li> <li>Internal reset by illegal-memory access</li> </ul>			
Power-on-reset circuit	<ul style="list-style-type: none"> <li>Power-on-reset: 1.51 V (TYP.)</li> <li>Power-down-reset: 1.50 V (TYP.)</li> </ul>			
Voltage detector	<ul style="list-style-type: none"> <li>Rising edge: 1.67 V to 4.06 V (14 steps)</li> <li>Falling edge: 1.63 V to 3.98 V (14 steps)</li> </ul>			
On-chip debug function	Provided			
Power supply voltage	$V_{DD} = 1.6$ to $5.5$ V (TA = -40 to +85°C) $V_{DD} = 2.4$ to $5.5$ V (TA = -40 to +105°C)			
Operating ambient temperature	Consumer applications: TA = -40 to +85°C Industrial applications: TA = -40 to +105°C			

**Notes 1.** The values in parentheses are the number of signal outputs when 8 com is used.

**2.** This reset occurs when instruction code FFH is executed.

This reset does not occur during emulation using an in-circuit emulator or an on-chip debugging emulator.

## CHAPTER 2 PIN FUNCTIONS

### 2.1 Port Function

The I/O buffer power supply for pins is provided by  $V_{DD}$ . Set in each port I/O, buffer, pull-up resistor is also valid for alternate functions.



## 2.1.1 64-pin products

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-5-10	I/O	Digital input invalid <sup>Note 1</sup>	SEG43/SO00/ TxD0/TOOLTxD	Port 0. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P03, P05, and P06 can be set to TTL input buffer. Output of P00 and P04 to P07 can be set to N-ch open-drain output ( $V_{DD}$ tolerance). Can be set to LCD output <sup>Note 2</sup> .
P01	7-5-4			(TI05)/(TO05)/INTP5/ PCLBUZ1/SEG44	
P02				INTP7/PCLBUZ0/ SEG45	
P03	8-5-4			RxD2/SEG46/VCOUT0	
P04	7-5-10			TxD2/SEG47/VCOUT1	
P05	8-5-10			SCK10/SCL10/SEG48	
P06				SI10/RxD1/SDA10/ SEG49	
P07	7-5-10			SO10/TxD1/ (PCLBUZ0)/SEG50	
P10	7-10-3	I/O	Analog input	ANI22/SEG35	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P16 and P17 can be set to TTL input buffer. Output of P16 and P17 can be set to N-ch open-drain output ( $V_{DD}$ tolerance). P10 to P13 can be set to analog input <sup>Note 3</sup> . P10 to P13 can be set to LCD output <sup>Note 3</sup> . P14 to P17 can be set to LCD output <sup>Note 2</sup> .
P11				ANI23/SEG36	
P12				ANI24/SEG37	
P13				ANI25/SEG38	
P14	7-5-4	Digital input invalid <sup>Note 1</sup>	TI04/TO04/SEG39	Ti07/TO07/SEG40	
P15				SEG41/SCK00/SCL00	
P16	8-5-10			SEG42/SI00/RxD0/ TOOLRxD/SDA00	
P17					
P20	4-3-3	I/O	Analog input	ANI11/AV <sub>REFM</sub>	Port 2. 5-bit I/O port. Input/output can be specified in 1-bit units. P22, P26, and P27 use of an on-chip pull-up resistor can be specified by a software setting at input port. P20 and P21 can be set to analog input <sup>Note 4</sup> . P22, P26, and P27 can be set to analog input <sup>Note 3</sup> . P22, P26, and P27 can be set to LCD output <sup>Note 3</sup> .
P21				ANI0/AV <sub>REFF</sub>	
P22	7-10-3			ANI16/SEG29	
P26				ANI20/SEG33	
P27				ANI21/SEG34	

**Notes 1.** “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

- Digital or LCD for each pin can be selected with the port mode register x (PMx) and the LCD port function register x (PFSEGx) (can be set in 1-bit unit).
- Digital, analog, or LCD for each pin can be selected with the port mode control register x (PMCx), the port mode register x (PMx), and the LCD port function register x (PFSEGx) (can be set in 1-bit unit).
- Setting digital or analog to each pin can be done in 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). See **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
P30	7-5-4	I/O	Digital input invalid <sup>Note 1</sup>	TI03/TO03/SEG20/ REMOOUT	Port 3. 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. Can be set to LCD output <sup>Note 2</sup> .
P31				INTP3/RTC1HZ/SEG21	
P32				TI01/TO01/SEG22	
P33				INTP4/SEG23	
<R> P40	7-1-3	I/O	Input port	TOOL0/(TI00)/(TO00)	Port 4. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P43 and P44 can be set to TTL input buffer. Output of P42 to P44 can be set to N-ch open-drain output ( $V_{DD}$ tolerance). P42 to P44 can be set to analog input <sup>Note 3</sup> . P45 can be set to analog input <sup>Note 4</sup> .
P42	7-6-4			TI05/TO05/(SO10)/ (TxD1)/IVREF1	
P43	8-6-4			(INTP7)/(SI10)/(RxD1)/ (SDA10)/IVCMP1	
P44				(SCK10)/(SCL10)/ IVCMP0	
P45	7-6-9		Digital input invalid <sup>Note 1</sup>	IVREF0	
P52	7-5-4	I/O	Digital input invalid <sup>Note 1</sup>	TI00/TO00/INTP1/ SEG6	Port 5. 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. Input of P53 can be set to TTL input buffer. Output of P53 and P54 can be set to N-ch open-drain output ( $V_{DD}$ tolerance). Can be set to LCD output <sup>Note 2</sup> .
P53	8-5-10			INTP2/SEG7	
P54	7-5-10			TI02/TO02/SEG8	
P57	7-5-4			INTP6/SEG11	
<R> P60	12-1-3	I/O	Input port	SCLA0/(TI01)/(TO01)	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. N-ch open-drain output (6 V tolerance).
P61				SDAA0/(TI02)/(TO02)	
P70	7-5-4	I/O	Digital input invalid <sup>Note 1</sup>	KR0/SEG12	Port 7. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Can be set to LCD output <sup>Note 2</sup> .
P74	7-5-16			KR4/SEG16/TKBO00	
P75				KR5/SEG17/TKBO01-2	
P76				KR6/SEG18/TKBO01-1	
P77				KR7/SEG19/TKBO01-0	

**Notes 1.** "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

- Digital or LCD for each pin can be selected with the port mode register x (PMx) and the LCD port function register x (PFSEGx) (can be set in 1-bit unit).
- Digital or analog for each pin can be selected with the port mode control register x (PMCx) (can be set in 1-bit unit).
- Digital or analog for each pin can be selected with the port mode control register x (PMCx) and the LCD port function register 3 (PFSEG3).

**Remark** Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). See **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
P121	2-2-1	Input	Input port	X1	Port 12. 3-bit I/O port and 4-bit input only port. For P125 to P127, input/output can be specified in 1-bit units. For P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port. P125 to P127 can be set to LCD output <sup>Note 2</sup> .
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P125	7-5-6	I/O	Digital input invalid <sup>Note 1</sup>	V <sub>L3</sub> /TI06/TO06	
P126	7-5-5			CAPL/(TI04)/(TO04)	
P127				CAPH/(TI03)/(TO03)/(REMOOUT)	
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input only port.
RESET	2-1-1	Input	–	–	Input only pin for external reset. When external reset is not used, connect this pin to V <sub>DD</sub> directly or via a resistor.

&lt;R&gt;

- Notes 1.** “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.
- 2.** Digital or LCD for each pin can be selected with the port mode register x (PMx) and the LCD port function register x (PFSEGx) (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). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

## 2.1.2 80-pin products

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-5-10	I/O	Digital input invalid <sup>Note 1</sup>	SEG43/SO00/ TxD0/TOOLTxD	Port 0. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P03, P05, and P06 can be set to TTL input buffer. Output of P00 and P04 to P07 can be set to N-ch open-drain output (V <sub>DD</sub> tolerance). Can be set to LCD output <sup>Note 2</sup> .
P01	7-5-4			(TI05)/(TO05)/(INTP5)/ PCLBUZ1/SEG44	
P02				INTP7/PCLBUZ0/ SEG45	
P03	8-5-4			RxD2/SEG46/VCOUT0	
P04	7-5-10			TxD2/SEG47/VCOUT1	
P05	8-5-10			SCK10/SCL10/SEG48	
P06				SI10/RxD1/SDA10/ SEG49	
P07	7-5-10			SO10/TxD1/ (PCLBUZ0)/SEG50	
P10	7-10-3	I/O	Analog input	ANI22/SEG35	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P16 and P17 can be set to TTL input buffer. Output of P16 and P17 can be set to N-ch open-drain output (V <sub>DD</sub> tolerance). P10 to P13 can be set to analog input <sup>Note 3</sup> . P10 to P13 can be set to LCD output <sup>Note 3</sup> . P14 to P17 can be set to LCD output <sup>Note 2</sup> .
P11				ANI23/SEG36	
P12				ANI24/SEG37	
P13				ANI25/SEG38	
P14	7-5-4		Digital input invalid <sup>Note 1</sup>	Ti04/TO04/SEG39	
P15				Ti07/TO07/SEG40	
P16	8-5-10		SEG41/SCK00/SCL00		
P17			SEG42/SI00/RxD0/ TOOLRxD/SDA00		
P20	4-3-3	I/O	Analog input	ANI11/AV <sub>REFM</sub>	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units. P22 to P27 use of an on-chip pull-up resistor can be specified by a software setting at input port. P20 and P21 can be set to analog input <sup>Note 4</sup> . P22 to P27 can be set to analog input <sup>Note 3</sup> . P22 to P27 can be set to LCD output <sup>Note 3</sup> .
P21				ANI0/AV <sub>REFP</sub>	
P22	7-10-3			ANI16/SEG29	
P23				ANI17/SEG30	
P24				ANI18/SEG31	
P25				ANI19/SEG32	
P26				ANI20/SEG33	
P27				ANI21/SEG34	

**Notes 1.** “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

- Digital or LCD for each pin can be selected with the port mode register x (PMx) and the LCD port function register x (PFSEGx) (can be set in 1-bit unit).
- Digital, analog, or LCD for each pin can be selected with the port mode control register x (PMCx), the port mode register x (PMx), and the LCD port function register x (PFSEGx) (can be set in 1-bit unit).
- Setting digital or analog to each pin can be done in 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). See **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
P30	7-5-4	I/O	Digital input invalid <sup>Note 1</sup>	TI03/TO03/SEG20/ REMOOUT	Port 3. 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. Input of P34 can be set to TTL input buffer. Output of P35 can be set to N-ch open-drain output ( $V_{DD}$ tolerance). Can be set to LCD output <sup>Note 2</sup> .
P31				INTP3/RTC1HZ/SEG21	
P32				TI01/TO01/SEG22	
P33				INTP4/SEG23	
P34				RxD3/SEG24	
P35				TxD3/SEG25	
P40	7-1-3	I/O	Input port	TOOL0/(TI00)/(TO00)	Port 4. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P43, P44, P46, and P47 can be set to TTL input buffer. Output of P42 to P44, P46, and P47 can be set to N-ch open-drain output ( $V_{DD}$ tolerance). P42 to P44 can be set to analog input <sup>Note 3</sup> . P45 can be set to analog input <sup>Note 4</sup> . P46 and P47 can be set to LCD output <sup>Note 2</sup> .
P41	7-6-4			(TI07)/(TO07)	
P42				TI05/TO05/(SO10)/ (TxD1)/IVREF1	
P43	8-6-4			(INTP7)/(SI10)/(RxD1)/ (SDA10)/IVCMP1	
P44	7-6-9		(SCK10)/(SCL10)/ IVCMP0		
P45			Digital input invalid <sup>Note 1</sup>	IVREF0	
P46			8-5-10	(SCK00)/(SCL00)/ SEG26	
P47			(SI00)/(RxD0)/ (SDA00)/SEG27		
P50	7-5-4	I/O	Digital input invalid <sup>Note 1</sup>	SEG4	Port 5. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P53 and P55 can be set to TTL input buffer. Output of P53 to P56 can be set to N-ch open-drain output ( $V_{DD}$ tolerance). Can be set to LCD output <sup>Note 2</sup> .
P51	7-5-10			SEG5	
P52				TI00/TO00/INTP1/ SEG6	
P53				8-5-10	
P54	7-5-10			TI02/TO02/SEG8	
P55	8-5-10			INTP5/SEG9	
P56	7-5-10			TI06/TO06/SEG10	
P57	7-5-4			INTP6/SEG11	

**Notes 1.** "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

- Digital or LCD for each pin can be selected with the port mode register x (PMx) and the LCD port function register x (PFSEGx) (can be set in 1-bit unit).
- Digital or analog for each pin can be selected with the port mode control register x (PMCx) (can be set in 1-bit unit).
- Digital or analog for each pin can be selected with the port mode control register x (PMCx) and the LCD port function register 3 (PFSEG3).

**Remark** Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). See **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
<R> P60	12-1-3	I/O	Input port	SCLA0/(TI01)/(TO01)	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. N-ch open-drain output (6 V tolerance).
P61				SDAA0/(TI02)/(TO02)	
P70	7-5-4	I/O	Digital input invalid <sup>Note 1</sup>	KR0/SEG12	Port 7. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Can be set to LCD output <sup>Note 2</sup> .
P71				KR1/SEG13	
P72				KR2/SEG14	
P73				KR3/SEG15	
P74	7-5-16			KR4/SEG16/TKBO00	
P75				KR5/SEG17/TKBO01-2	
P76				KR6/SEG18/TKBO01-1	
P77				KR7/SEG19/TKBO01-0	
P121	2-2-1	Input	Input port	X1	Port 12. 3-bit I/O port and 4-bit input only port. For P125 to P127, input/output can be specified in 1-bit units. For P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port. P125 to P127 can be set to LCD output <sup>Note 2</sup> .
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P125	7-5-6	I/O	Digital input invalid <sup>Note 1</sup>	V <sub>L3</sub> /(TI06)/(TO06)	
P126	7-5-5			CAPL/(TI04)/(TO04)	
P127				CAPH/(TI03)/(TO03)/(REMOOUT)	
P130	7-5-10	I/O	Digital input invalid <sup>Note 1</sup>	(SO00)/(TxD0)/SEG28	Port 13. 1-bit I/O port and 1-bit input only port. For P130, input/output can be specified. For P130, use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P130 can be set to N-ch open-drain output (V <sub>DD</sub> tolerance). P130 can be set to LCD output <sup>Note 2</sup> .
P137	2-1-2	Input	Input port	INTP0	
RESET	2-1-1	Input	–	–	Input only pin for external reset. When external reset is not used, connect this pin to V <sub>DD</sub> directly or via a resistor.

**Notes 1.** “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

- Digital or LCD for each pin can be selected with the port mode register x (PMx) and the LCD port function register x (PFSEGx) (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). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

## 2.2 Functions Other than Port Pins

### 2.2.1 With functions for each product

(1/2)

Function Name	80-pin	64-pin	Function Name	80-pin	64-pin	Function Name	80-pin	64-pin
ANI0	√	√	REMOOUT	√	√	TKBO00	√	√
ANI1	√	√	RESET	√	√	TKBO01-0	√	√
ANI16	√	√	RxD0	√	√	TKBO01-1	√	√
ANI17	√	–	RxD1	√	√	TKBO01-2	√	√
ANI18	√	–	RxD2	√	√	VL1	√	√
ANI19	√	–	RxD3	√	–	VL2	√	√
ANI20	√	√	TxD0	√	√	VL3	√	√
ANI21	√	√	TxD1	√	√	VL4	√	√
ANI22	√	√	TxD2	√	√	CAPH	√	√
ANI23	√	√	TxD3	√	–	CAPL	√	√
ANI24	√	√	SCK00	√	√	X1	√	√
ANI25	√	√	SCK10	√	√	X2	√	√
INTP0	√	√	SI00	√	√	EXCLK	√	√
INTP1	√	√	SI10	√	√	XT1	√	√
INTP2	√	√	SO00	√	√	XT2	√	√
INTP3	√	√	SO10	√	√	EXCLKS	√	√
INTP4	√	√	SCL00	√	√	VDD	√	√
INTP5	√	√	SCL10	√	√	AVREFF	√	√
INTP6	√	√	SDA00	√	√	AVREFM	√	√
INTP7	√	√	SDA10	√	√	VSS	√	√
IVCMP0	√	√	SDAA0	√	√	TOOLRxD	√	√
IVCMP1	√	√	SCLA0	√	√	TOOLTxD	√	√
IVREF0	√	√	TI00	√	√	TOOL0	√	√
IVREF1	√	√	TI01	√	√	COM0	√	√
VCOUT0	√	√	TI02	√	√	COM1	√	√
VCOUT1	√	√	TI03	√	√	COM2	√	√
KR0	√	√	TI04	√	√	COM3	√	√
KR1	√	–	TI05	√	√	COM4	√	√
KR2	√	–	TI06	√	√	COM5	√	√
KR3	√	–	TI07	√	√	COM6	√	√
KR4	√	√	TO00	√	√	COM7	√	√
KR5	√	√	TO01	√	√	SEG0	√	√
KR6	√	√	TO02	√	√	SEG1	√	√
KR7	√	√	TO03	√	√	SEG2	√	√
PCLBUZ0	√	√	TO04	√	√	SEG3	√	√
PCLBUZ1	√	√	TO05	√	√	SEG4	√	–
RTC1HZ	√	√	TO06	√	√	SEG5	√	–
REGC	√	√	TO07	√	√	SEG6	√	√

(2/2)

Function Name	80-pin	64-pin	Function Name	80-pin	64-pin	Function Name	80-pin	64-pin
SEG7	√	√	SEG22	√	√	SEG37	√	√
SEG8	√	√	SEG23	√	√	SEG38	√	√
SEG9	√	–	SEG24	√	–	SEG39	√	√
SEG10	√	–	SEG25	√	–	SEG40	√	√
SEG11	√	√	SEG26	√	–	SEG41	√	√
SEG12	√	√	SEG27	√	–	SEG42	√	√
SEG13	√	–	SEG28	√	–	SEG43	√	√
SEG14	√	–	SEG29	√	√	SEG44	√	√
SEG15	√	–	SEG30	√	–	SEG45	√	√
SEG16	√	√	SEG31	√	–	SEG46	√	√
SEG17	√	√	SEG32	√	–	SEG47	√	√
SEG18	√	√	SEG33	√	√	SEG48	√	√
SEG19	√	√	SEG34	√	√	SEG49	√	√
SEG20	√	√	SEG35	√	√	SEG50	√	√
SEG21	√	√	SEG36	√	√			



## 2.2.2 Description of Functions

(1/2)

Function Name	I/O	Function
ANI0, ANI1, ANI16 to ANI25	Input	A/D converter analog input (see <b>Figure 12-38 Analog Input Pin Connection</b> )
INTP0 to INTP7	Input	External interrupt request input Specified the valid edge: Rising edge, falling edge, or both rising and falling edges
IVCMP0, IVCMP1	Input	Comparator analog voltage input
IVREF0, IVREF1	Input	Comparator reference voltage input
VCOU0, VCOU1	Output	Comparator output
KR0 to KR7	Input	Key interrupt input
PCLBUZ0, PCLBUZ1	Output	Clock output/buzzer output
REGC	–	Pin for connecting regulator output stabilization capacitance for internal operation. Connect this pin to $V_{SS}$ via a capacitor (0.47 to 1 $\mu$ F). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.
REMOOUT	Output	Remote controller output
RTC1HZ	Output	Real-time clock 2 correction clock (1 Hz) output
$\overline{\text{RESET}}$	Input	This is the active-low system reset input pin. When the external reset pin is not used, connect this pin directly or via a resistor to $V_{DD}$ .
RxD0 to RxD3	Input	Serial data input pins of serial interface UART0 to UART3
TxD0 to TxD3	Output	Serial data output pins of serial interface UART0 to UART3
SCK00, SCK10	I/O	Serial clock I/O pins of serial interface CSI00 and CSI10
SI00, SI10	Input	Serial data input pins of serial interface CSI00 and CSI10
SO00, SO10	Output	Serial data output pins of serial interface CSI00 and CSI10
SCL00, SCL10	Output	Serial clock output pins of serial interface IIC00 and IIC10
SDA00, SDA10	I/O	Serial data I/O pins of serial interface IIC00 and IIC10
SCLA0	I/O	Serial clock I/O pins of serial interface IICA0
SDAA0	I/O	Serial data I/O pins of serial interface IICA0
TI00 to TI07	Input	The pins for inputting an external count clock/capture trigger to 16-bit timers 00 to 07
TO00 to TO07	Output	Timer output pins of 16-bit timers 00 to 07
TKBO00, TKBO01-0, TKBO01-1, TKBO01-2	Output	Timer output pins of 16-bit timer KB20
$V_{L1}$ to $V_{L4}$	–	LCD drive voltage
CAPH, CAPL	–	Connecting a capacitor for LCD controller/driver
X1, X2	–	Resonator connection for main system clock
EXCLK	Input	External clock input for main system clock
XT1, XT2	–	Resonator connection for subsystem clock
EXCLKS	Input	External clock input for subsystem clock

(2/2)

Function Name	I/O	Function
V <sub>DD</sub>	–	Positive power supply for all pins
AV <sub>REFP</sub>	Input	A/D converter reference potential (+ side) input
AV <sub>REFM</sub>	Input	A/D converter reference potential (– side) input
V <sub>SS</sub>	–	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
COM0 to COM7	Output	LCD controller/driver common signal outputs
SEG0 to SEG50	Output	LCD controller/driver segment signal outputs

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

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

P40/TOOL0	Operating Mode
V <sub>DD</sub>	Normal operation mode
0 V	Flash memory programming mode

For details, see 28.4 Serial Programming Method.

**Remark** Use bypass capacitors (about 0.1  $\mu$ F) as noise and latch up countermeasures with relatively thick wires at the shortest distance to V<sub>DD</sub> to V<sub>SS</sub> lines.

## 2.3 Connection of Unused Pins

Table 2-2 shows the connections of unused pins.

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

**Table 2-2. Connection of Unused Pins (1/3)**

Pin Name	I/O	Recommended Connection of Unused Pins
P00/SEG43/SO00/TxD0/TOOLTxD	I/O	<When setting to port I/O> Input: Independently connect to V <sub>DD</sub> or V <sub>SS</sub> via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P01/(TI05)/(TO05)/(INTP5)/PCLBUZ1/SEG44		
P02/INTP7/PCLBUZ0/SEG45		
P03/RxD2/SEG46/VCOU0		
P04/TxD2/SEG47/VCOU1		
P05/SCK10/SCL10/SEG48		
P06/SI10/RxD1/SDA10/SEG49		
P07/SO10/TxD1/(PCLBUZ0)/SEG50		
P10/ANI22/SEG35		<When setting to port I/O> Input: Independently connect to V <sub>DD</sub> or V <sub>SS</sub> via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P11/ANI23/SEG36		
P12/ANI24/SEG37		
P13/ANI25/SEG38		
P14/TI04/TO04/SEG39		
P15/TI07/TO07/SEG40		
P16/SEG41/SCK00/SCL00		
P17/SEG42/SI00/RxD0/TOOLRxD/SDA00		
P20/ANI1/AV <sub>REFM</sub>		Input: Independently connect to V <sub>DD</sub> or V <sub>SS</sub> via a resistor. Output: Leave open.
P21/ANI0/AV <sub>REFP</sub>		
P22/ANI16/SEG29	<When setting to port I/O> Input: Independently connect to V <sub>DD</sub> or V <sub>SS</sub> via a resistor. Output: Leave open. <When setting to segment output> Leave open.	
P23/ANI17/SEG30		
P24/ANI18/SEG31		
P25/ANI19/SEG32		
P26/ANI20/SEG33		
P27/ANI21/SEG34		

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

Table 2-2. Connection of Unused Pins (2/3)

Pin Name	I/O	Recommended Connection of Unused Pins
P30/TI03/TO03/SEG20/REMOOUT	I/O	<When setting to port I/O> Input: Independently connect to V <sub>DD</sub> or V <sub>SS</sub> via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P31/INTP3/RTC1HZ/SEG21		
P32/TI01/TO01/SEG22		
P33/INTP4/SEG23		
P34/RxD3/SEG24		
P35/TxD3/SEG25		
P40/TOOL0/(TI00)/(TO00)		
P41/(TI07)/(TO07)		
P42/TI05/TO05/(SO10)/(TxD1)/IVREF1		
P43/(INTP7)/(SI10)/(RxD1)/(SDA10)/IVCMP1		
P44/(SCK10)/(SCL10)/IVCMP0		
P45/IVREF0		
P46/(SCK00)/(SCL00)/SEG26		
P47/(SI00)/(RxD0)/(SDA00)/SEG27		
P50/SEG4		
P51/SEG5	I/O	<When setting to port I/O> Input: Independently connect to V <sub>DD</sub> or V <sub>SS</sub> via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P52/TI00/TO00/INTP1/SEG6		
P53/INTP2/SEG7		
P54/TI02/TO02/SEG8		
P55/INTP5/SEG9		
P56/TI06/TO06/SEG10		
P57/INTP6/SEG11		
P60/SCLA0/(TI01)/(TO01)		
P61/SDAA0/(TI02)/(TO02)		
P70/KR0/SEG12		
P71/KR1/SEG13		
P72/KR2/SEG14		
P73/KR3/SEG15		
P74/KR4/SEG16/TKBO00		
P75/KR5/SEG17/TKBO01-2		
P76/KR6/SEG18/TKBO01-1		
P77/KR7/SEG19/TKBO01-0		

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

Table 2-2. Connection of Unused Pins (3/3)

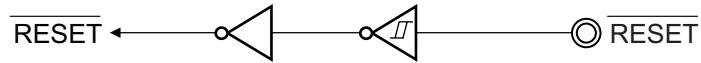
Pin Name	I/O	Recommended Connection of Unused Pins	
P121/X1	Input	Independently connect to V <sub>DD</sub> or V <sub>SS</sub> via a resistor.	
P122/X2/EXCLK			
P123/XT1			
P124/XT2/EXCLKS			
P125/V <sub>L3</sub> (TI06)/(TO06)	I/O	Input: Independently connect to V <sub>DD</sub> or V <sub>SS</sub> via a resistor. Output: Leave open.	
P126/CAPL(TI04)/(TO04)			
P127/CAPH(TI03)/(TO03)/(REMOOUT)		<When setting to port I/O> Input: Independently connect to V <sub>DD</sub> or V <sub>SS</sub> via a resistor. Output: Leave open. <When setting to segment output> Leave open.	
P130/(SO00)/(TxD0)/SEG28			
P137/INTP0	Input	Independently connect to V <sub>DD</sub> or V <sub>SS</sub> via a resistor.	
RESET	Input	Connect directly or via a resistor to V <sub>DD</sub> .	
REGC	–	Connect to V <sub>SS</sub> via capacitor (0.47 to 1 μF).	
COM0 to COM3	Output	Leave open.	
COM4/SEG0			
COM5/SEG1			
COM6/SEG2			
COM7/SEG3			
V <sub>L1</sub>			–
V <sub>L2</sub>			–
V <sub>L4</sub>			–

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

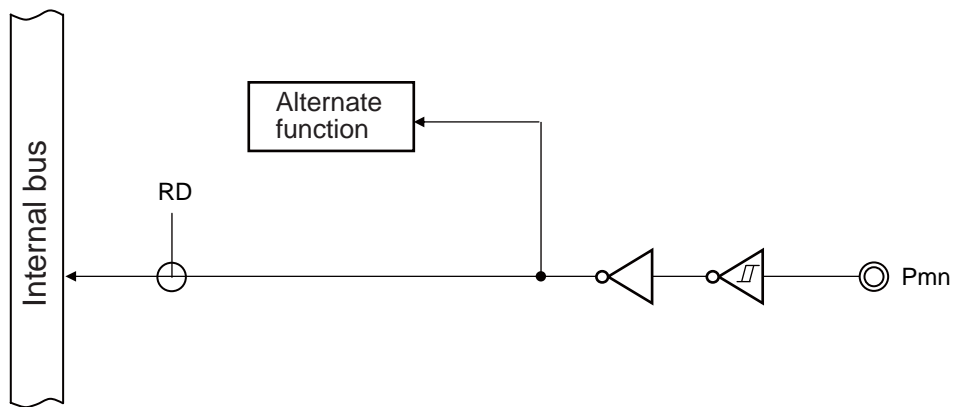
**2.4 Block Diagrams of Pins**

Figures 2-1 to 2-16 show the block diagrams of the pins described in **2.1.1 64-pin products** and **2.1.2 80-pin products**.

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

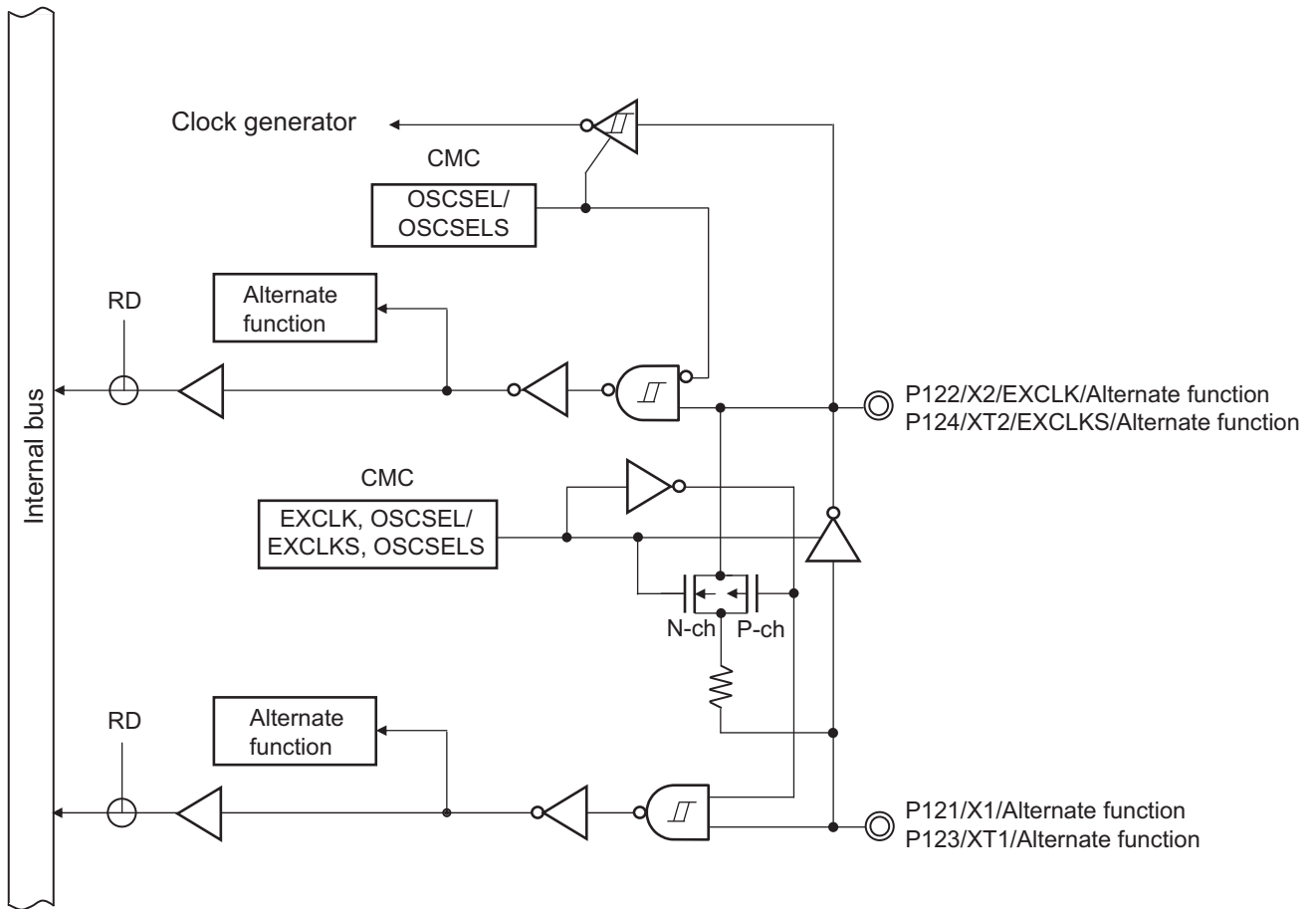


**Figure 2-2. Pin Block Diagram for Pin Type 2-1-2**



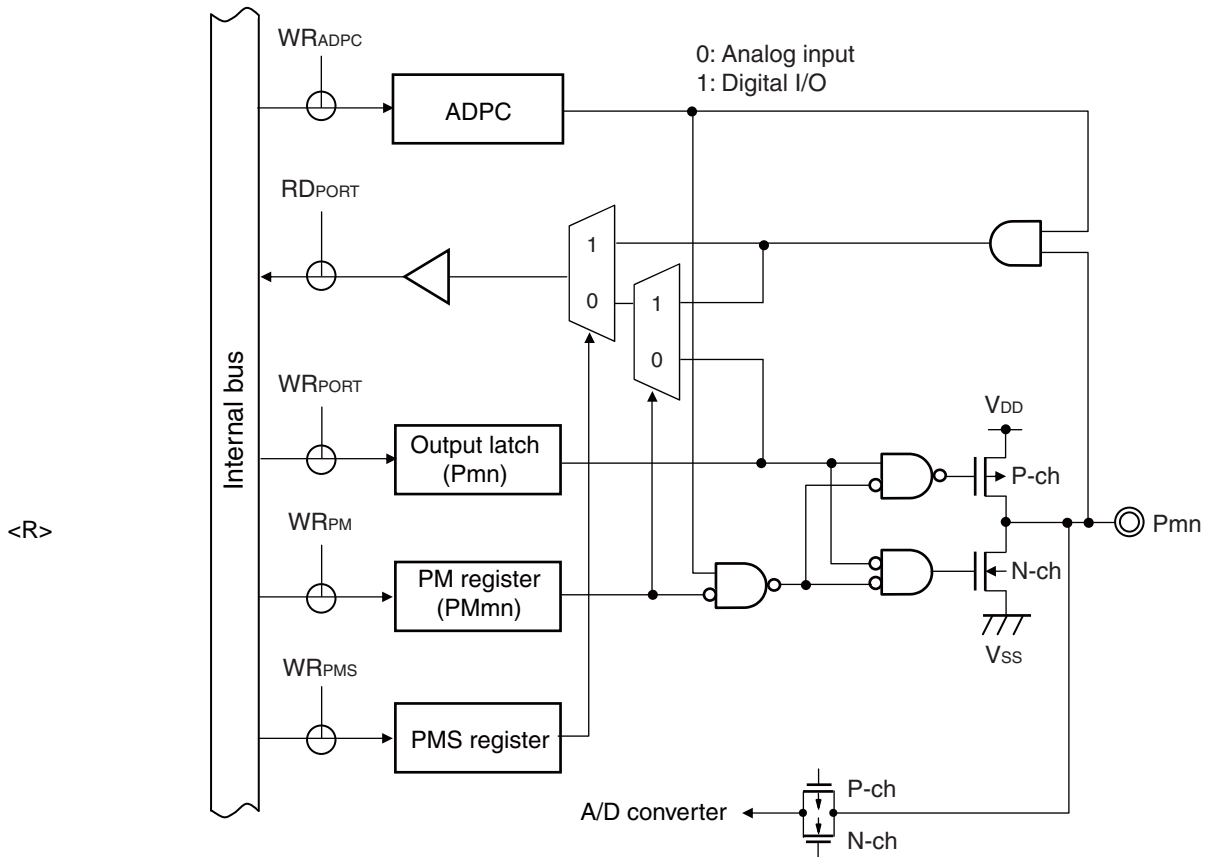
**Remark** For alternate functions, see **2.1 Port Function**.

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



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

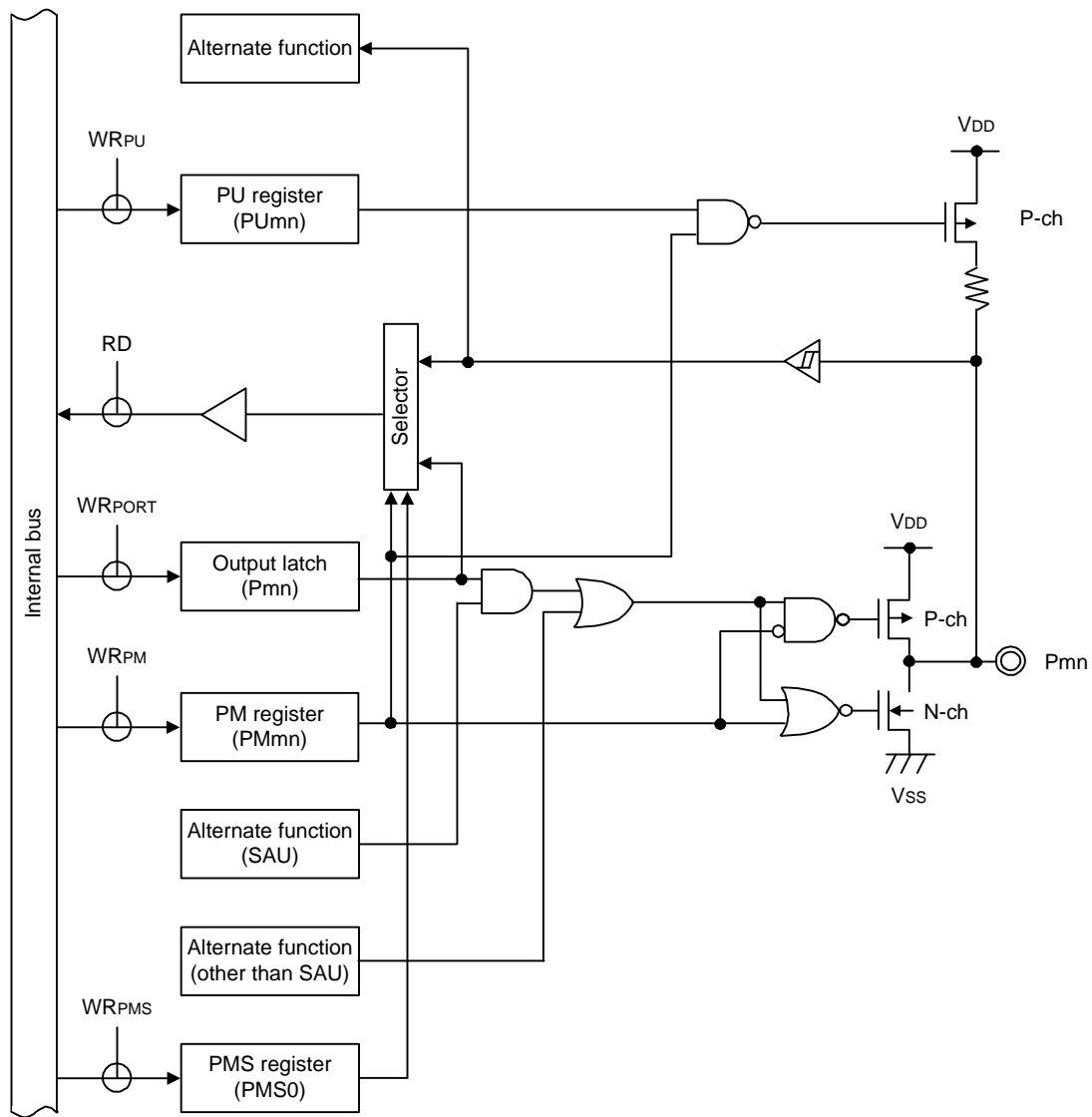
Figure 2-4. Pin Block Diagram for Pin Type 4-3-3





<R>

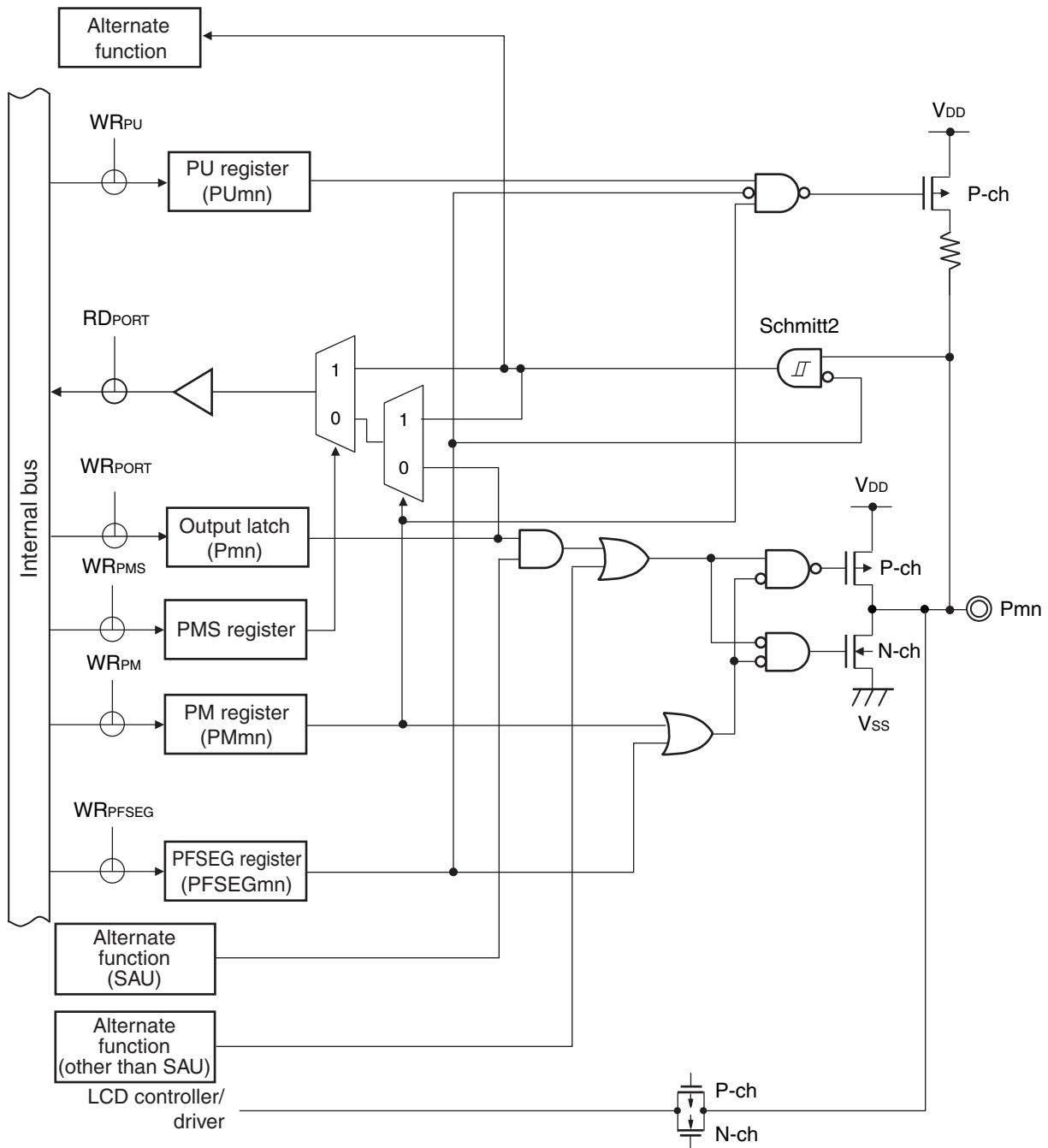
Figure 2-5. Pin Block Diagram for Pin Type 7-1-3



- Remarks**
1. For alternate functions, see 2.1 Port Function.
  2. SAU: Serial array unit

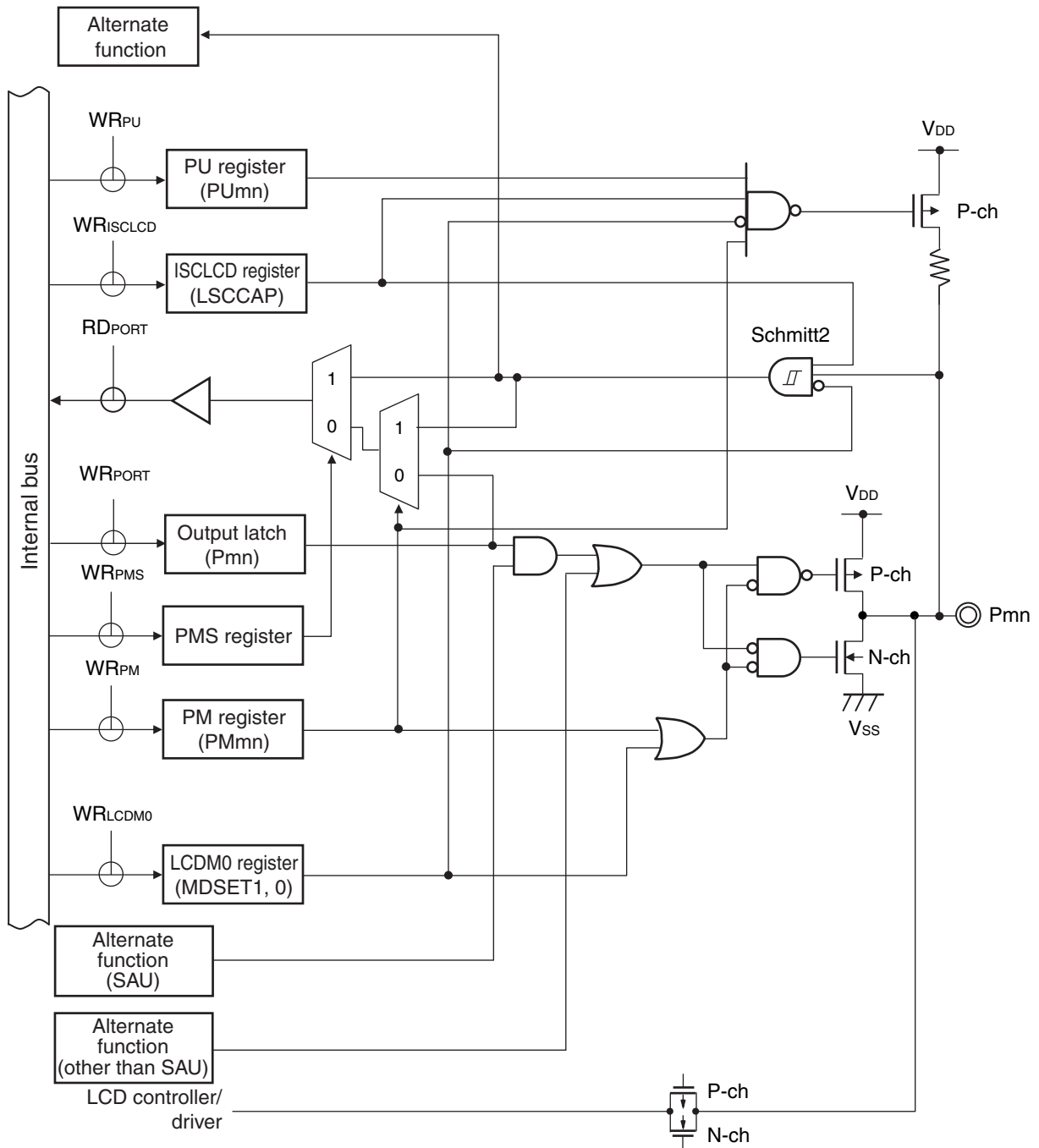
v

Figure 2-6. Pin Block Diagram for Pin Type 7-5-4



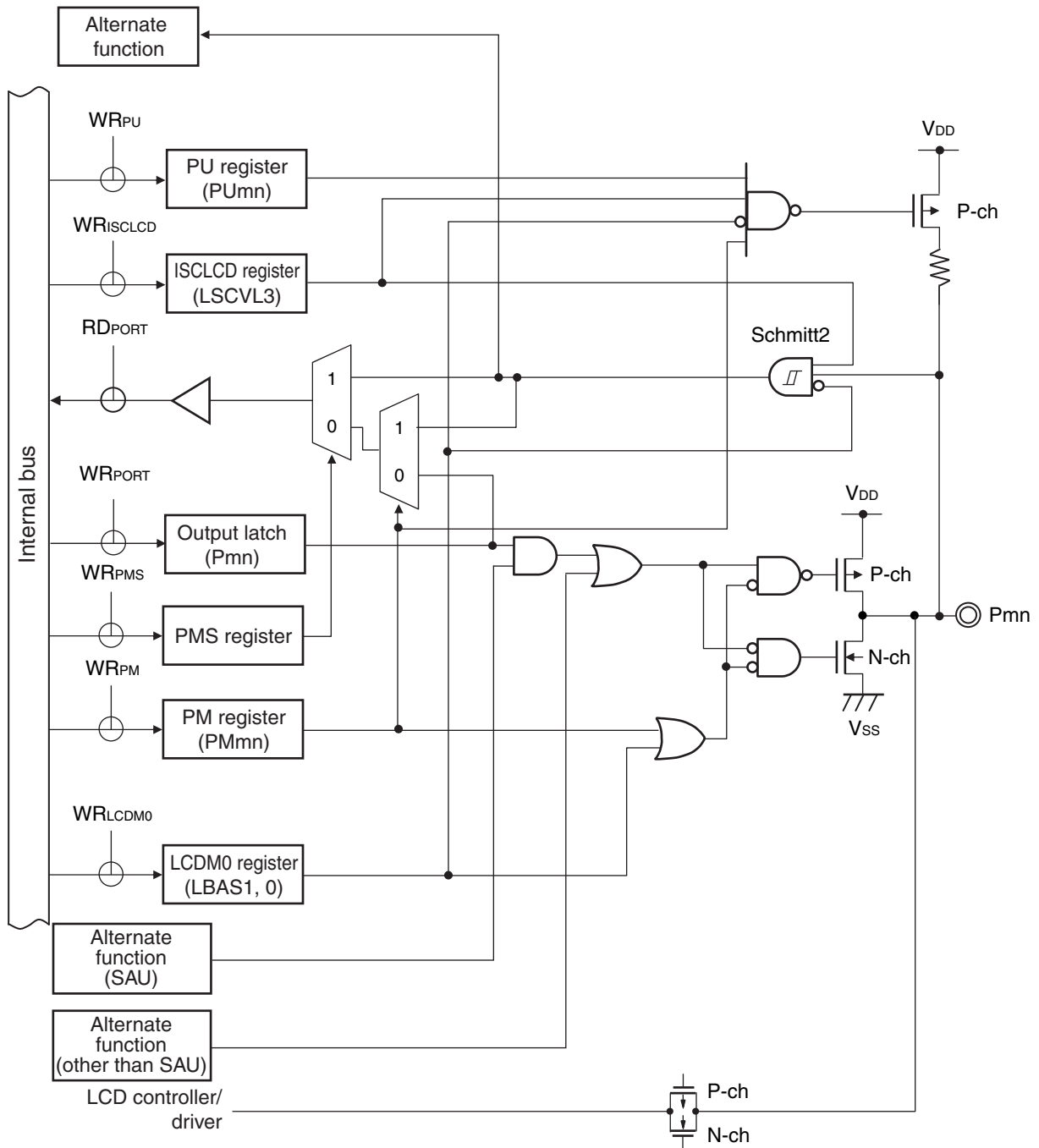
- Remarks**
1. For alternate functions, see 2.1 Port Function.
  2. SAU: Serial array unit

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



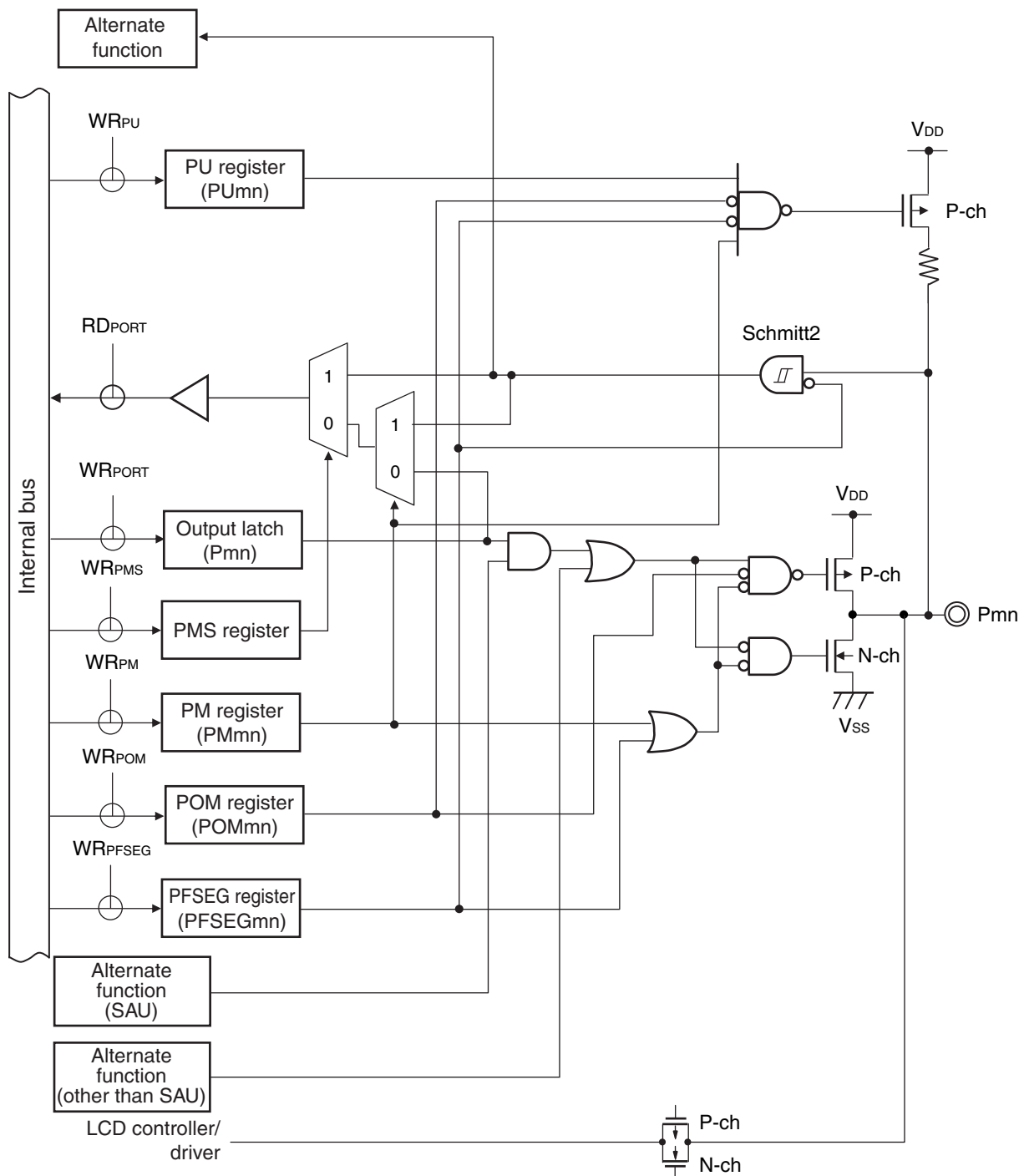
- Remarks**
1. For alternate functions, see 2.1 Port Function.
  2. SAU: Serial array unit

Figure 2-8. Pin Block Diagram for Pin Type 7-5-6



- Remarks**
1. For alternate functions, see 2.1 Port Function.
  2. SAU: Serial array unit

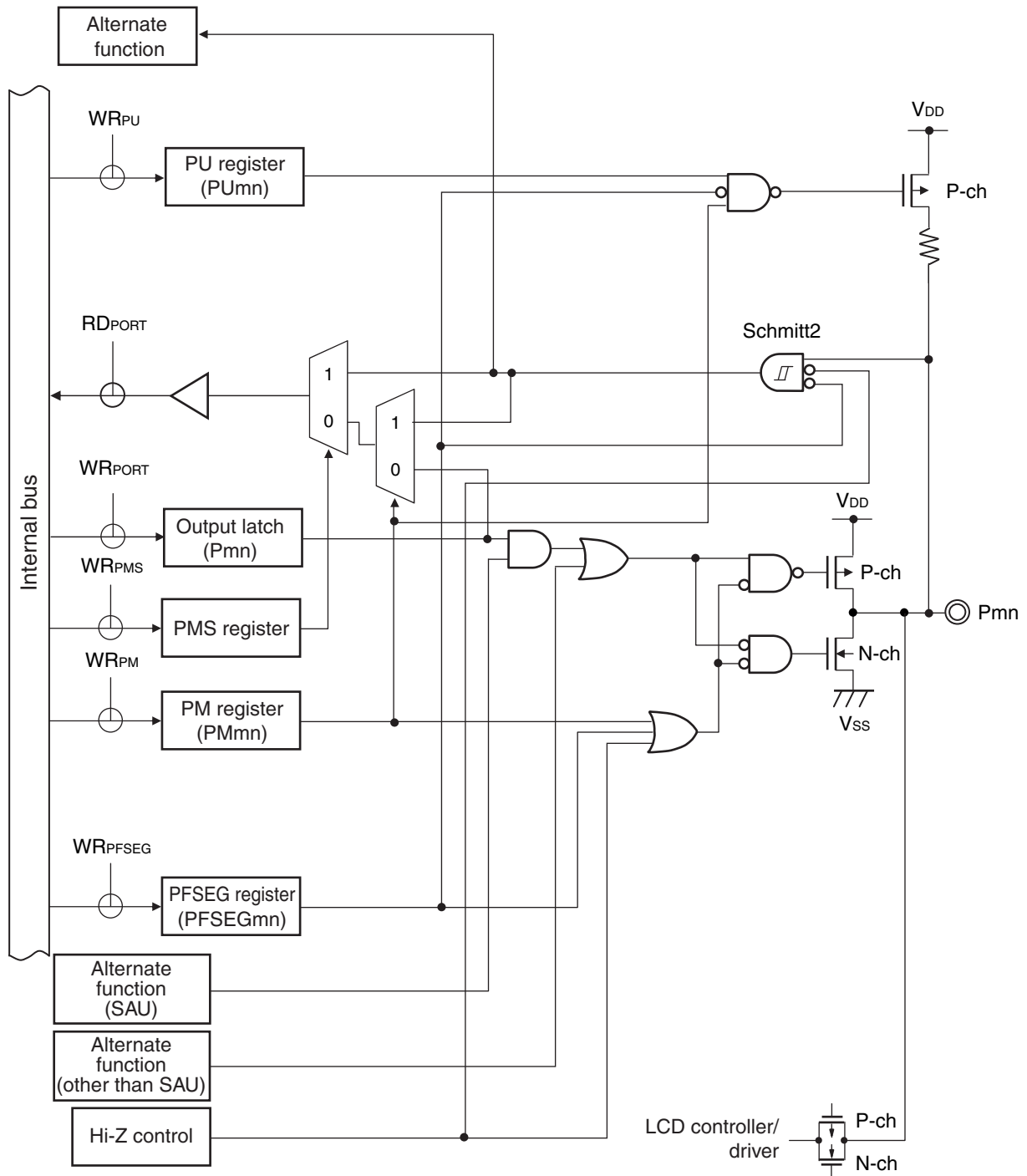
Figure 2-9. Pin Block Diagram for Pin Type 7-5-10



<R> **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 (POM<sub>x</sub>).

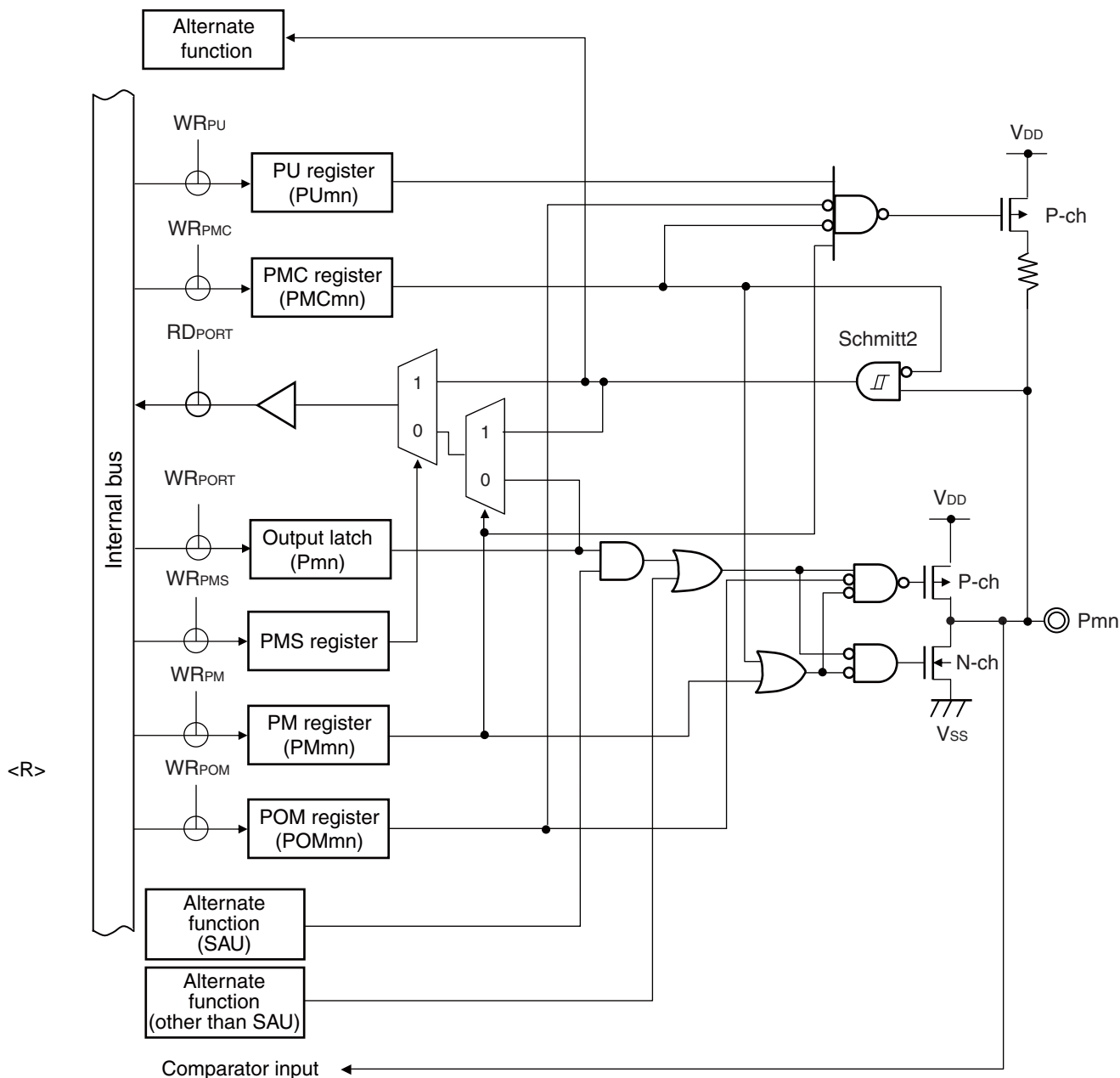
- Remarks**
1. For alternate functions, see 2.1 Port Function.
  2. SAU: Serial array unit

Figure 2-10. Pin Block Diagram for Pin Type 7-5-16



- Remarks**
1. For alternate functions, see 2.1 Port Function.
  2. SAU: Serial array unit

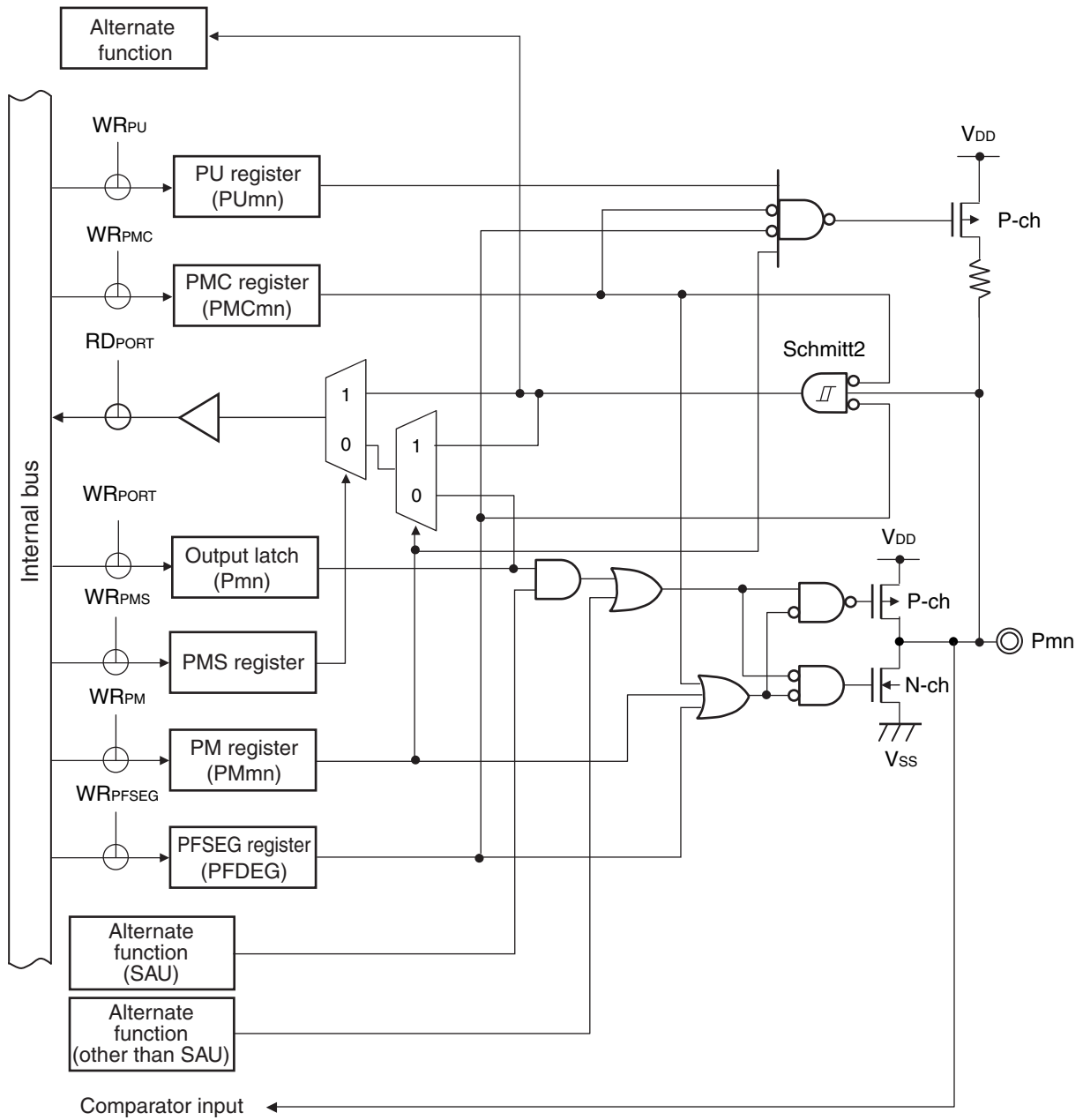
Figure 2-11. Pin Block Diagram for Pin Type 7-6-4



<R> **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

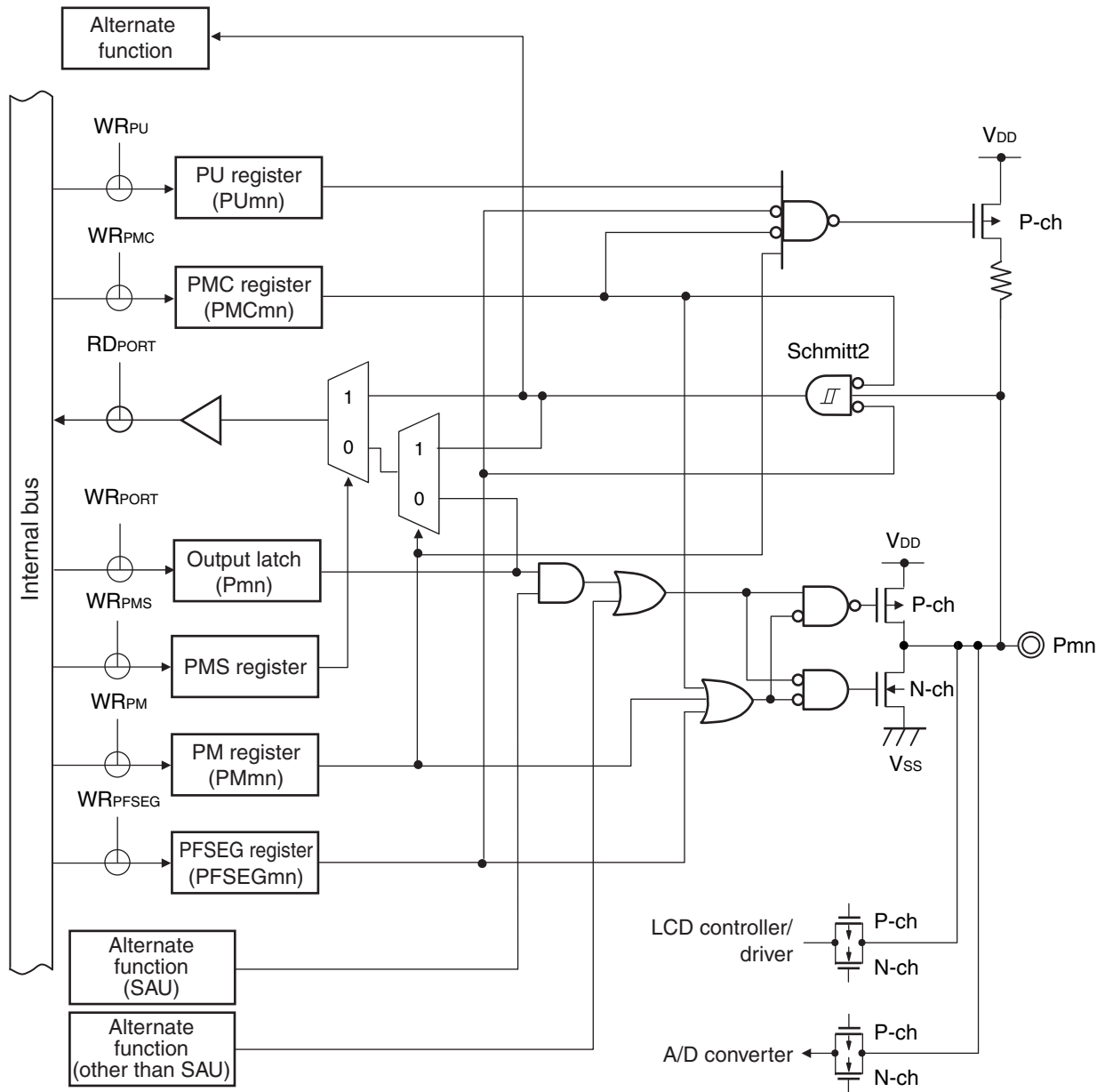
Figure 2-12. Pin Block Diagram for Pin Type 7-6-9



- Remarks**
1. For alternate functions, see 2.1 Port Function.
  2. SAU: Serial array unit

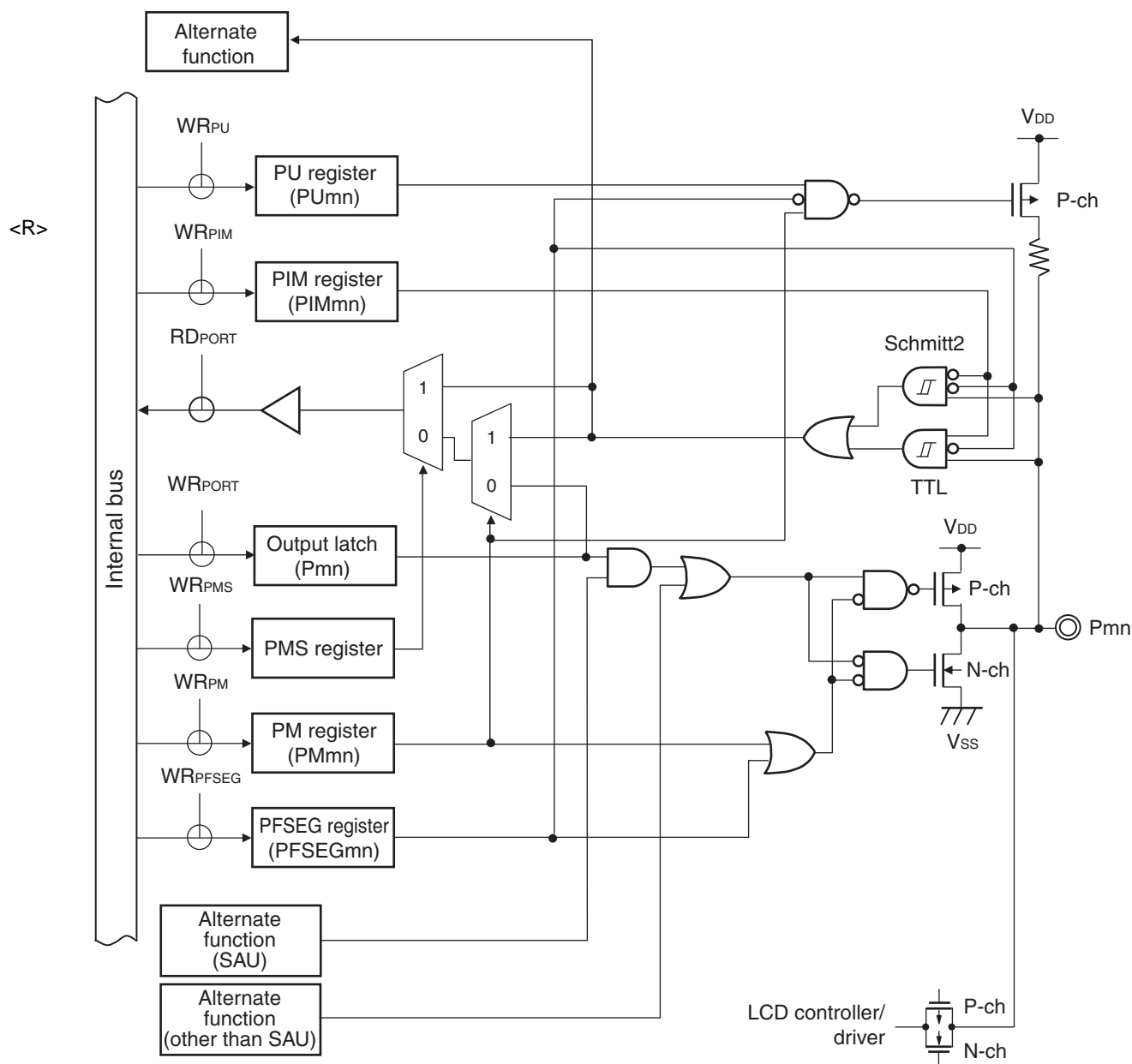


Figure 2-13. Pin Block Diagram for Pin Type 7-10-3



- Remarks**
1. For alternate functions, see 2.1 Port Function.
  2. SAU: Serial array unit

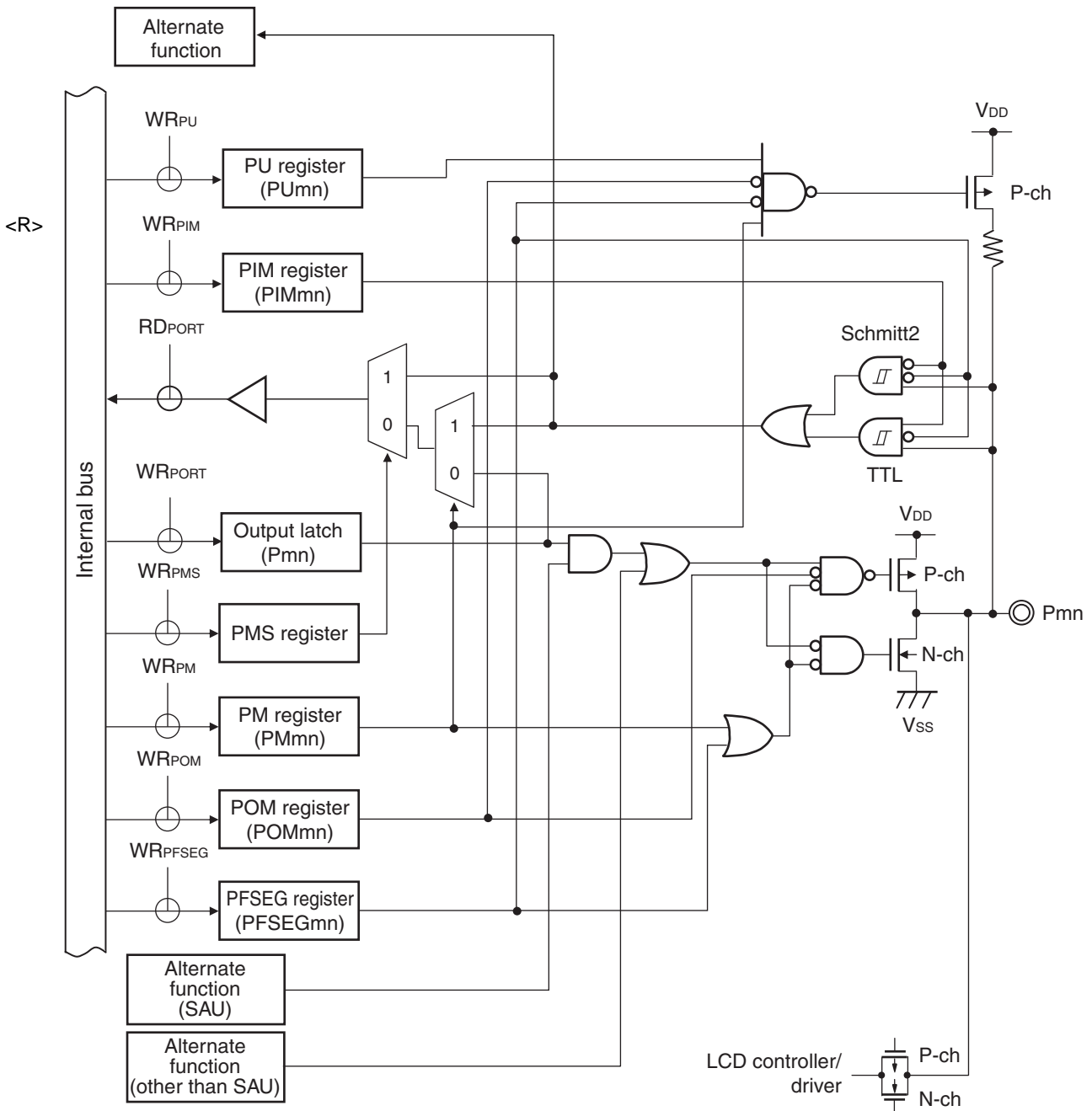
Figure 2-14. Pin Block Diagram for Pin Type 8-5-4



<R> **Caution** 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

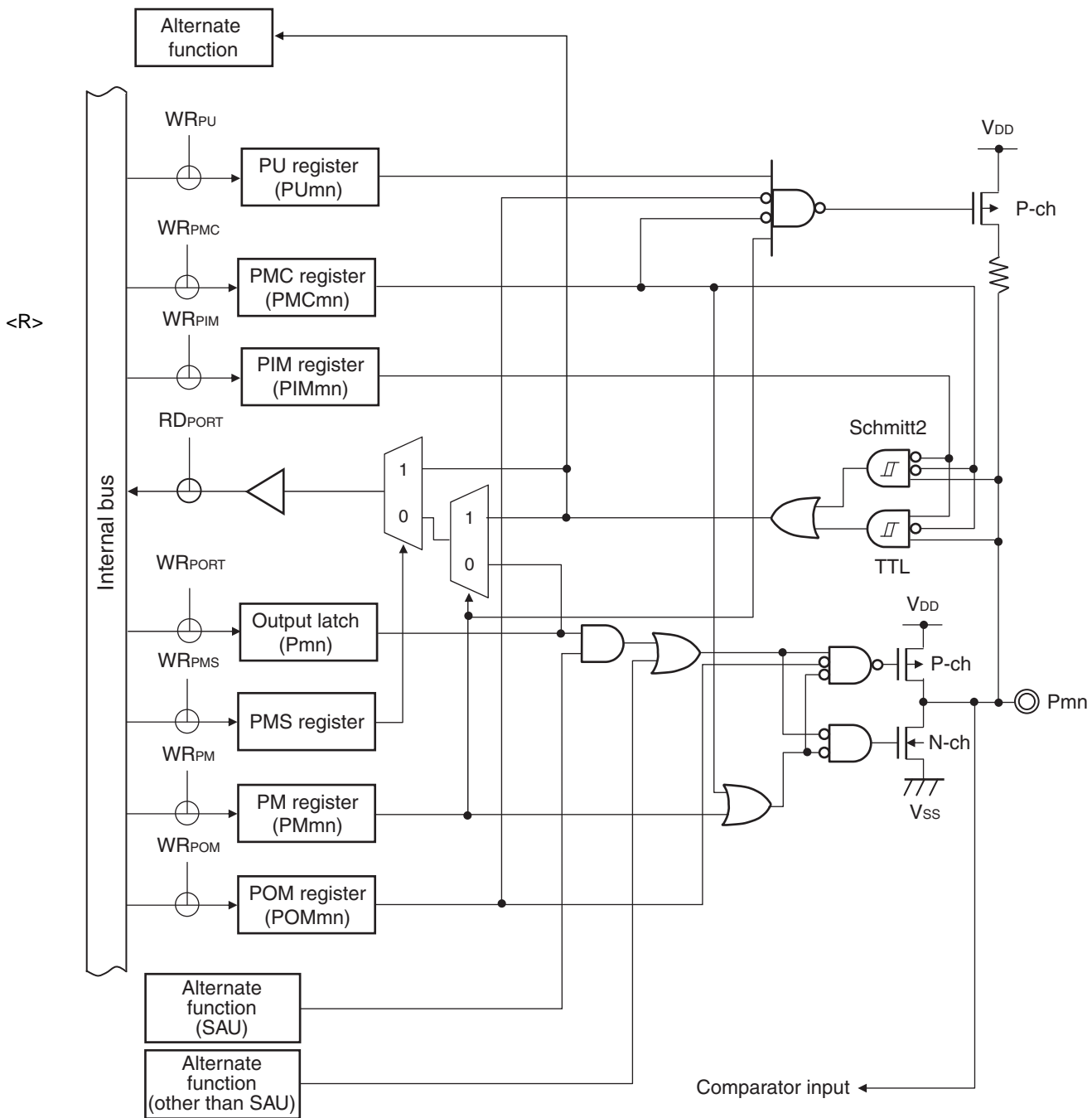
Figure 2-15. Pin Block Diagram for Pin Type 8-5-10



- <R> **Caution1** 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 2-16. Pin Block Diagram for Pin Type 8-6-4

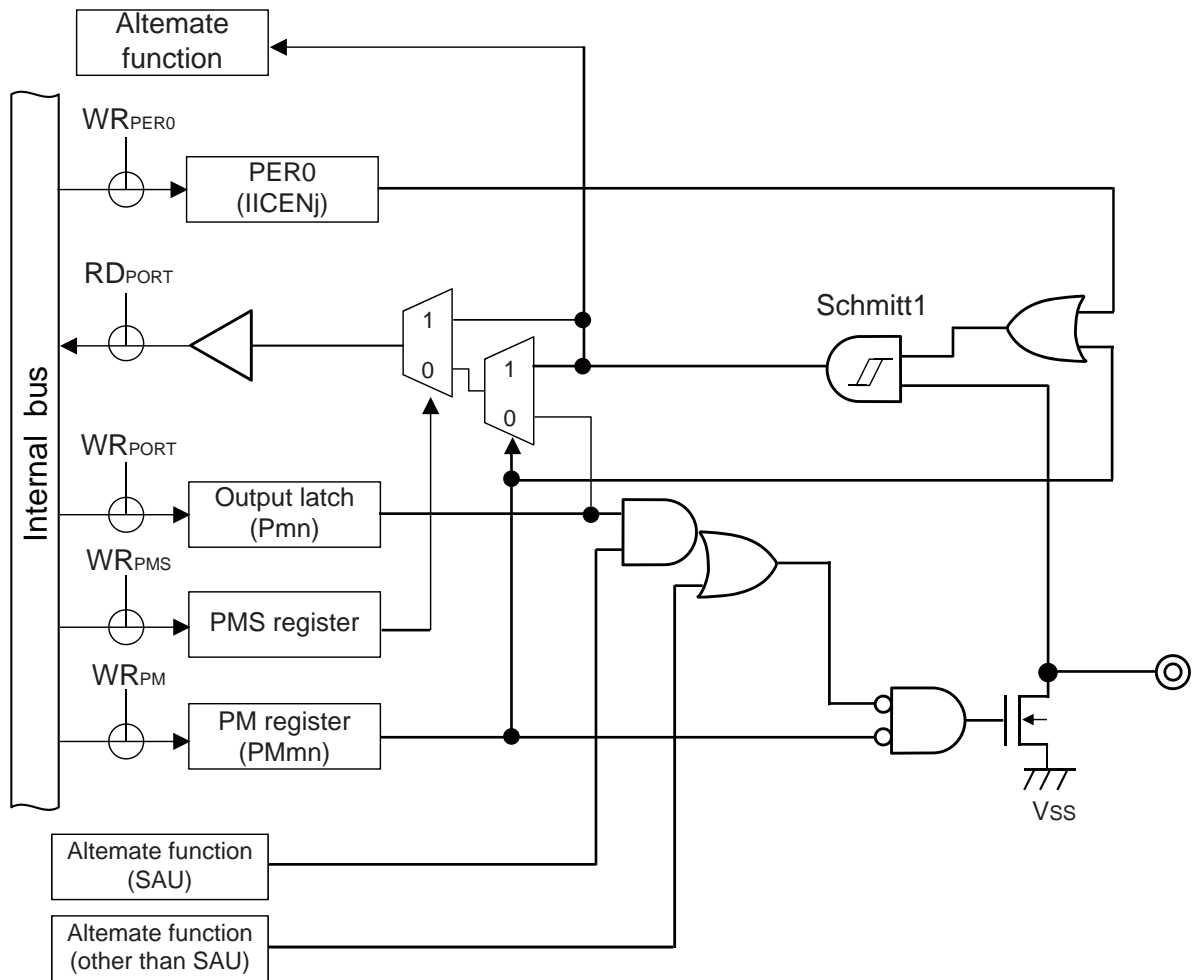


- <R> **Caution1** A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).
- 2** Because of TTL input buffer structure, if the port input mode register (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

<R>

Figure 2-17. Pin Block Diagram for Pin Type 12-1-3



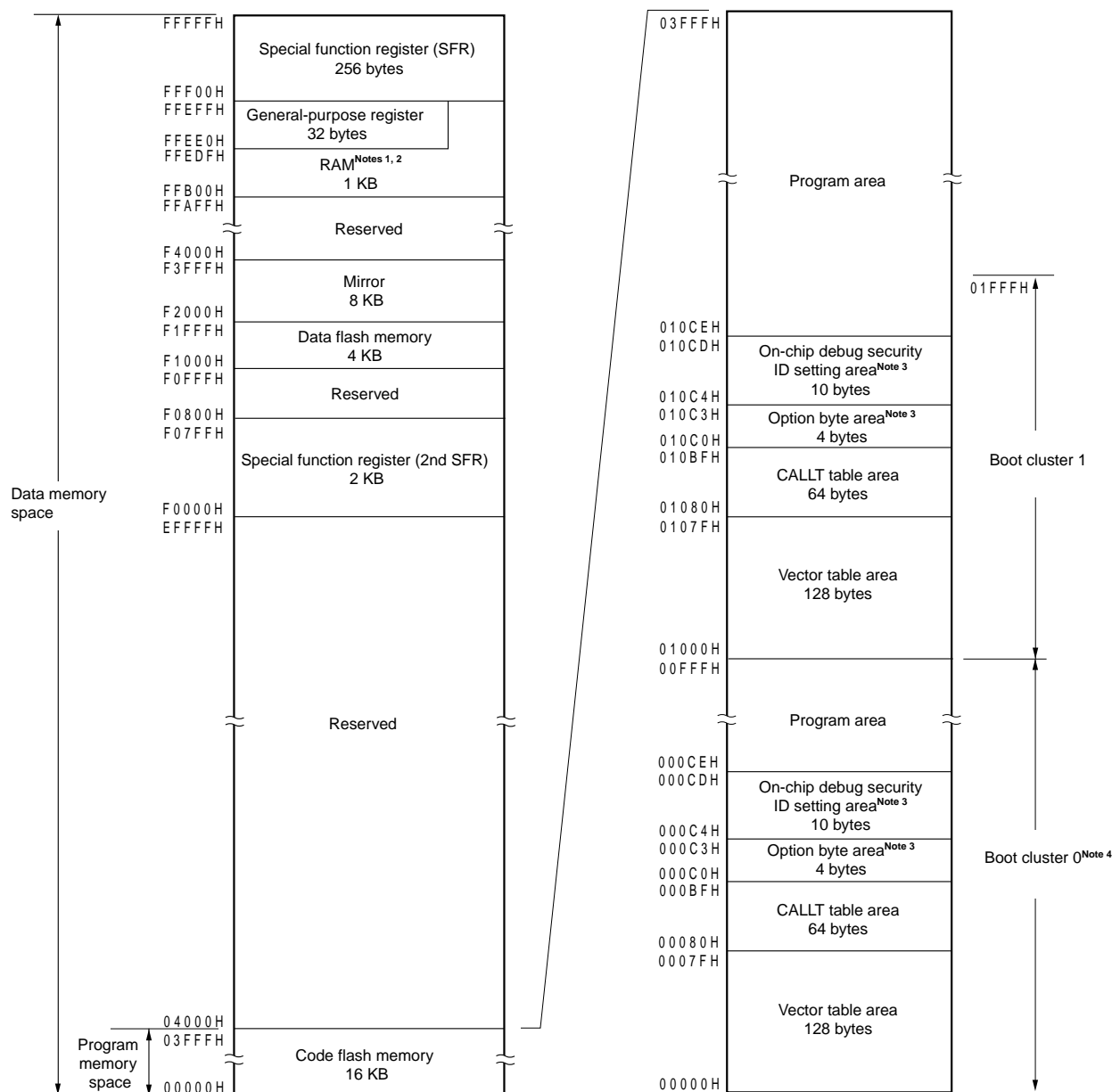
- Remarks**
1. For alternate functions, see 2.1 Port Function.
  2. SAU: Serial array unit

## CHAPTER 3 CPU ARCHITECTURE

### 3.1 Memory Space

Products in the RL78/L13 can access a 1 MB memory space. Figures 3-1 to 3-6 show the memory maps.

Figure 3-1. Memory Map (R5F10WLA, R5F10WMA)

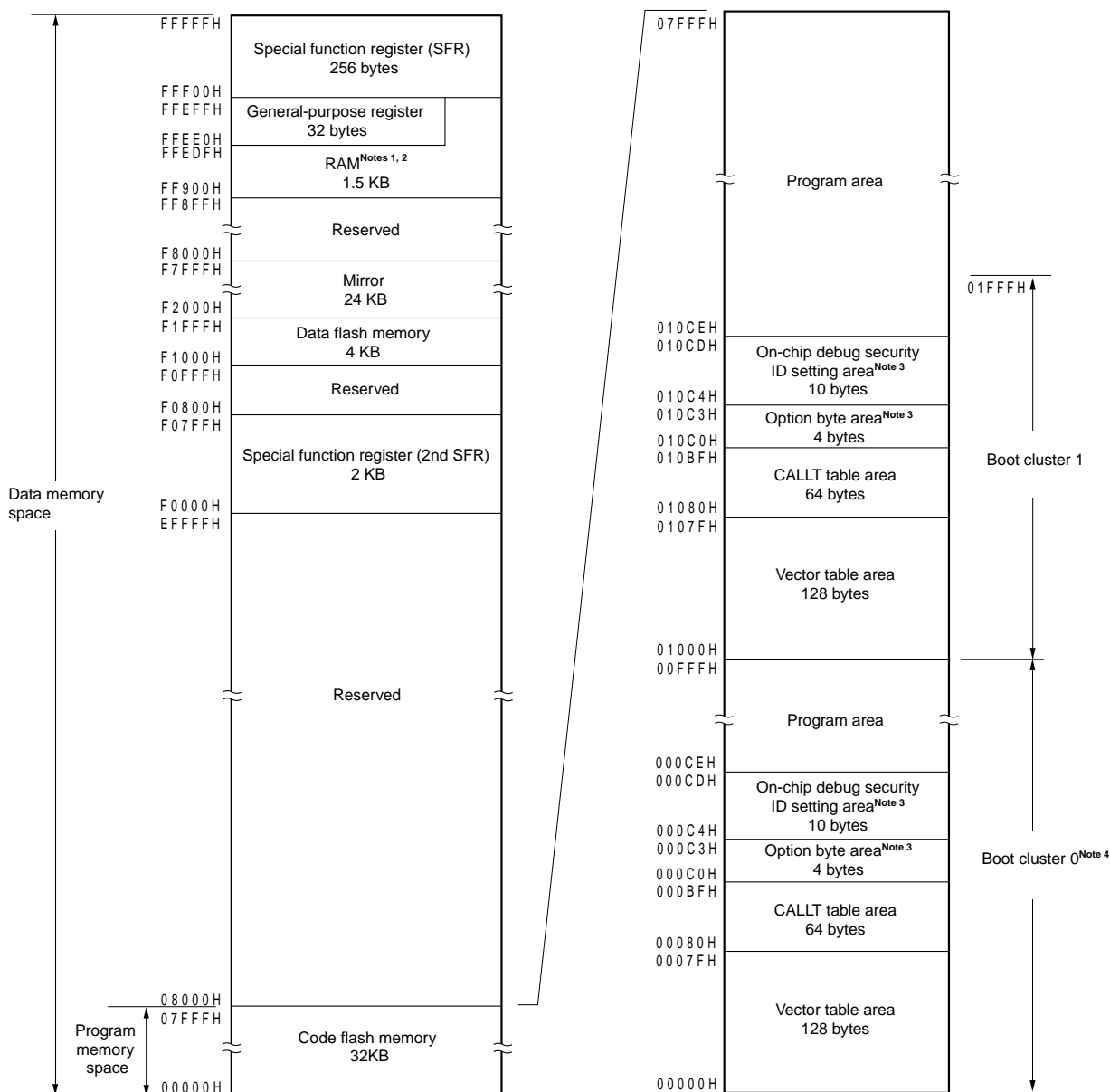


- Notes**
- Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DMA transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.
  - Instructions can be executed from the RAM area excluding the general-purpose register area.
  - 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.
  - Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.6 Security Setting**).

**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 **25.3.3 RAM parity error detection function**.

Figure 3-2. Memory Map (R5F10WLC, R5F10WMC)



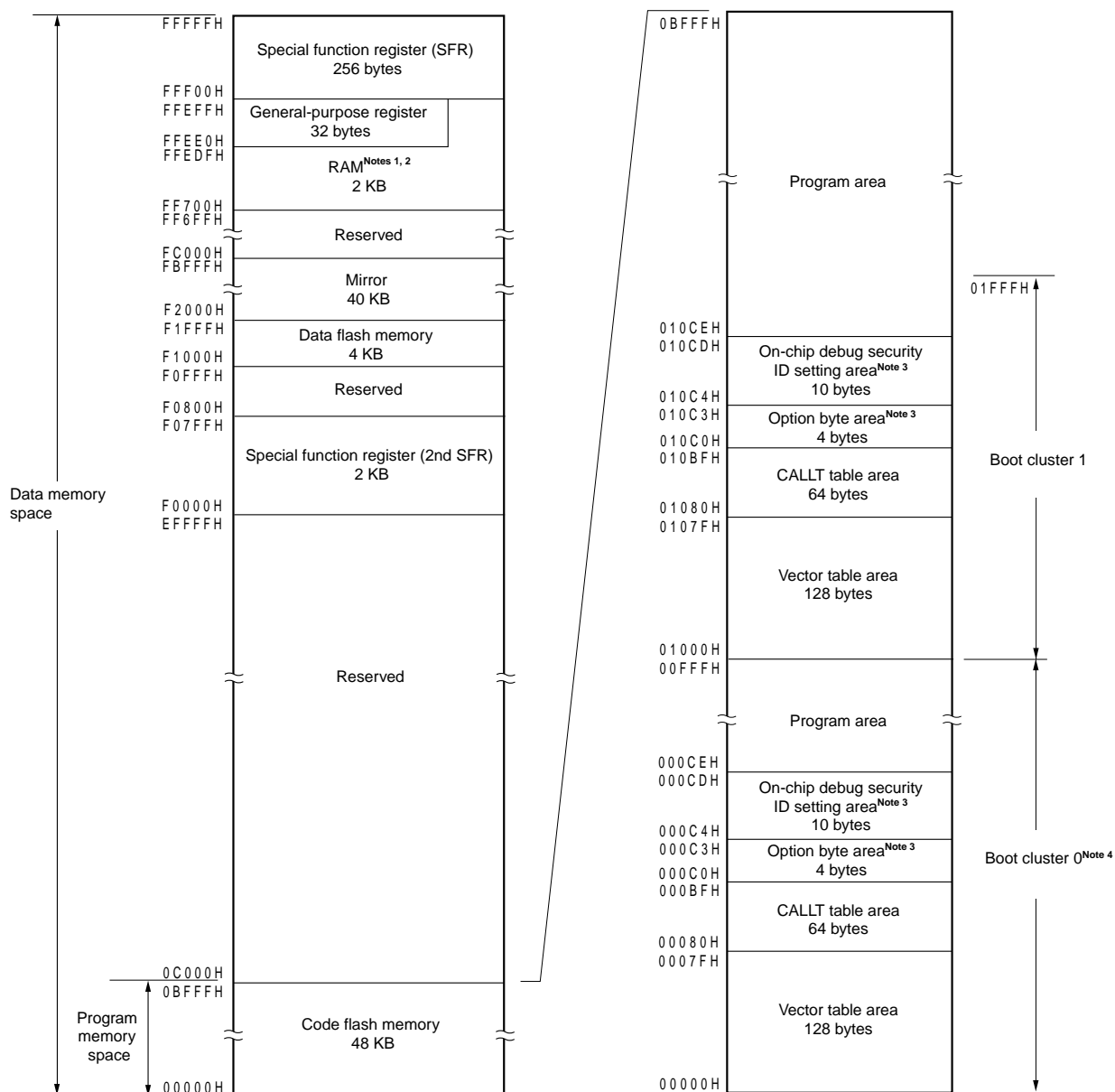
- Notes**
- Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DMA transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.
  - Instructions can be executed from the RAM area excluding the general-purpose register area.
  - 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.
  - Writing boot cluster 0 can be prohibited depending on the setting of security (see 28.6 Security Setting).

**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 25.3.3 RAM parity error detection function.



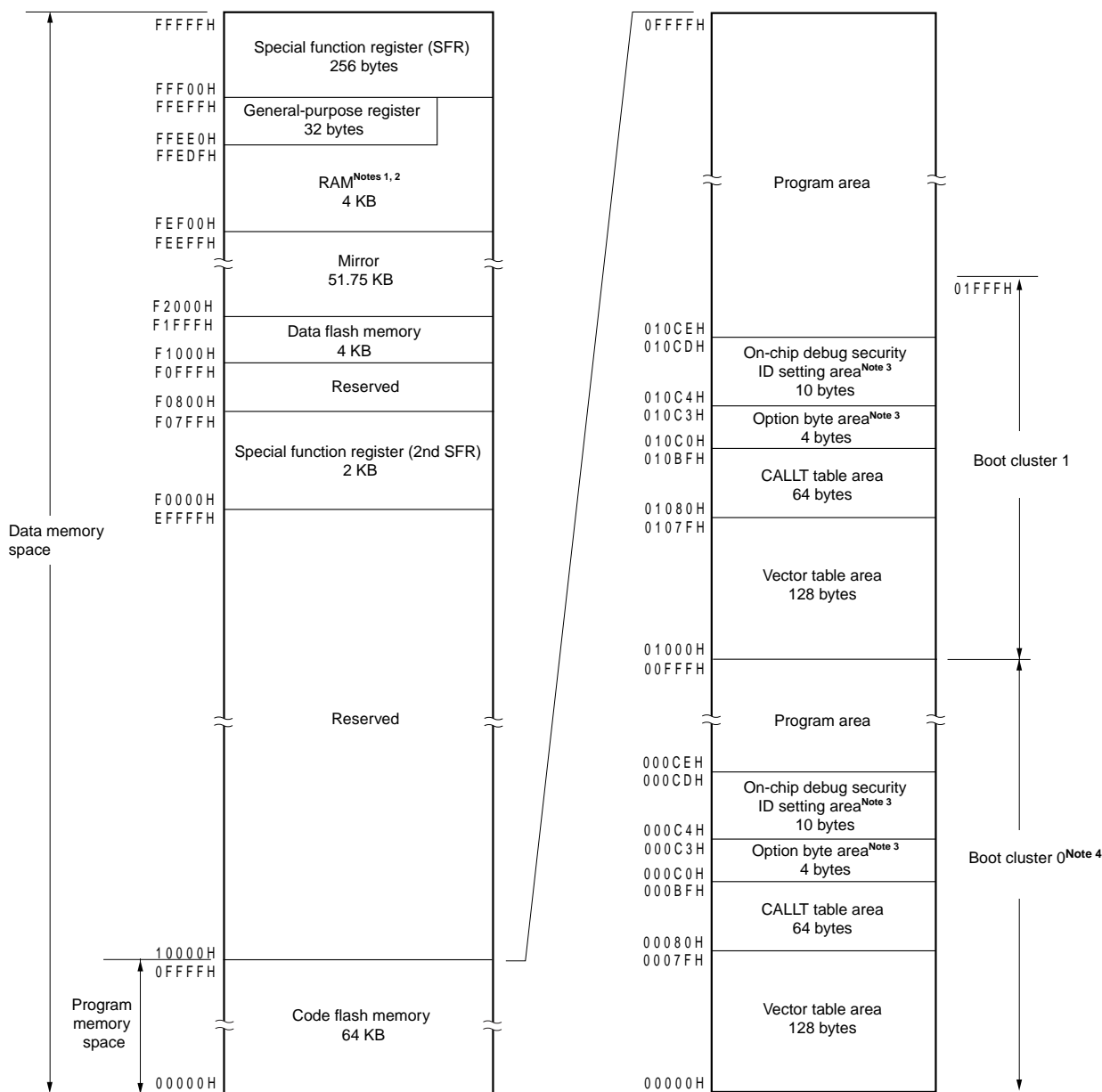
Figure 3-3. Memory Map (R5F10WLD, R5F10WMD)



- Notes**
- 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.
  - Instructions can be executed from the RAM area excluding the general-purpose register area.
  - 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.
  - Writing boot cluster 0 can be prohibited depending on the setting of security (see 28.6 Security Setting).

**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 25.3.3 RAM parity error detection function.

Figure 3-4. Memory Map (R5F10WLE, R5F10WME)

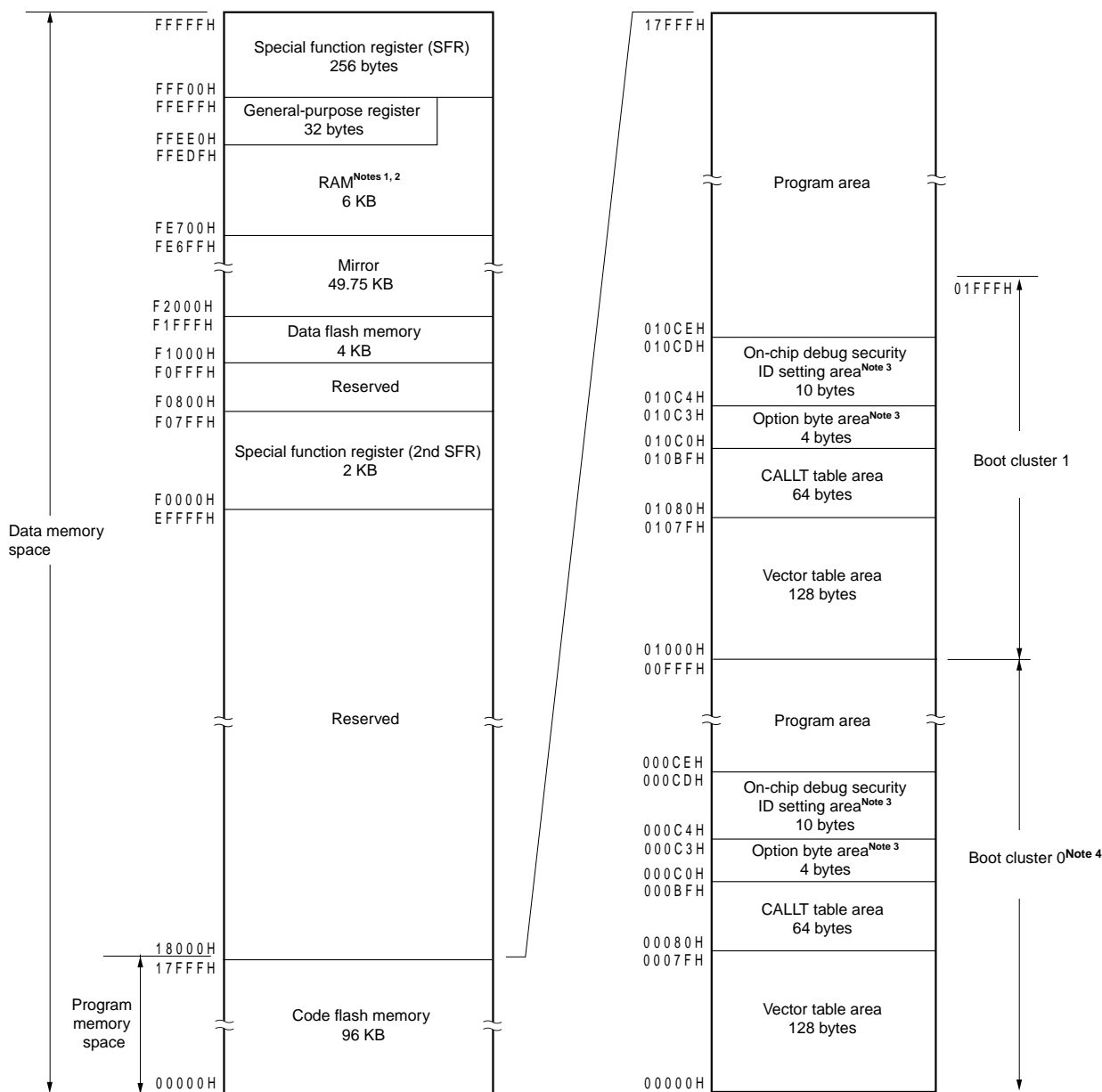


- Notes**
- Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DMA transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.
  - Instructions can be executed from the RAM area excluding the general-purpose register area.
  - 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.
  - Writing boot cluster 0 can be prohibited depending on the setting of security (see 28.6 Security Setting).

**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 25.3.3 RAM parity error detection function.

Figure 3-5. Memory Map (R5F10WLF, R5F10WMF)

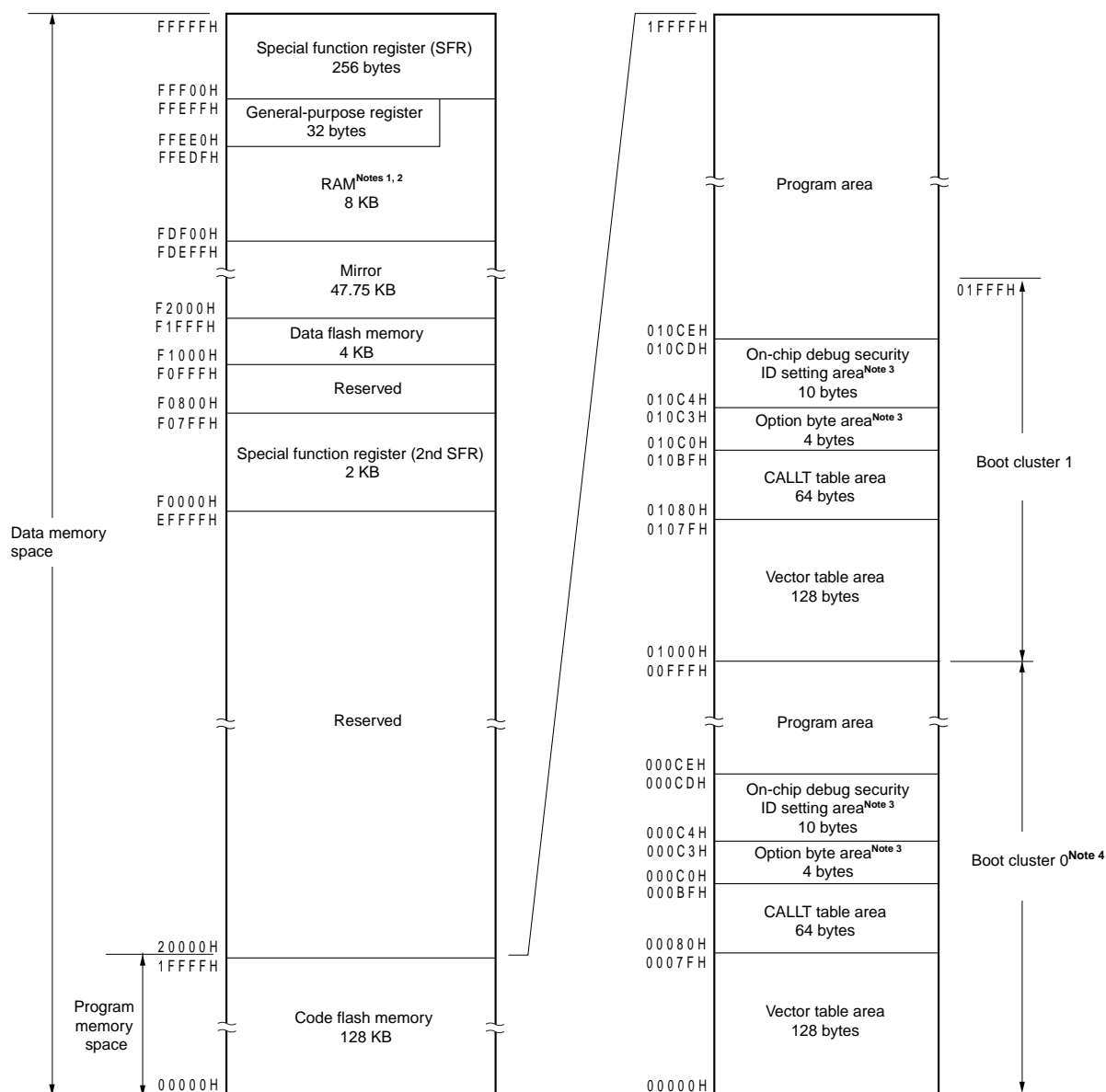


- Notes**
- Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DMA transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.
  - Instructions can be executed from the RAM area excluding the general-purpose register area.
  - 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.
  - Writing boot cluster 0 can be prohibited depending on the setting of security (see 28.6 Security Setting).

**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 25.3.3 RAM parity error detection function.

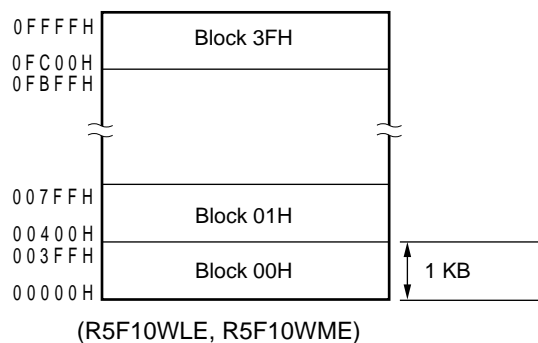
Figure 3-6. Memory Map (R5F10WLG, R5F10WMG)



- Notes**
- 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. Also, use of the area FDF00H to FE309H is prohibited, because this area is used for each library.
  - Instructions can be executed from the RAM area excluding the general-purpose register area.
  - 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.
  - Writing boot cluster 0 can be prohibited depending on the setting of security (see 28.6 Security Setting).

**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 25.3.3 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	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
00000H to 003FFH	00H	08000H to 083FFH	20H	10000H to 103FFH	40H	18000H to 183FFH	60H
00400H to 007FFH	01H	08400H to 087FFH	21H	10400H to 107FFH	41H	18400H to 187FFH	61H
00800H to 00BFFH	02H	08800H to 08BFFH	22H	10800H to 10BFFH	42H	18800H to 18BFFH	62H
00C00H to 00FFFH	03H	08C00H to 08FFFH	23H	10C00H to 10FFFH	43H	18C00H to 18FFFH	63H
01000H to 013FFH	04H	09000H to 093FFH	24H	11000H to 113FFH	44H	19000H to 193FFH	64H
01400H to 017FFH	05H	09400H to 097FFH	25H	11400H to 117FFH	45H	19400H to 197FFH	65H
01800H to 01BFFH	06H	09800H to 09BFFH	26H	11800H to 11BFFH	46H	19800H to 19BFFH	66H
01C00H to 01FFFH	07H	09C00H to 09FFFH	27H	11C00H to 11FFFH	47H	19C00H to 19FFFH	67H
02000H to 023FFH	08H	0A000H to 0A3FFH	28H	12000H to 123FFH	48H	1A000H to 1A3FFH	68H
02400H to 027FFH	09H	0A400H to 0A7FFH	29H	12400H to 127FFH	49H	1A400H to 1A7FFH	69H
02800H to 02BFFH	0AH	0A800H to 0ABFFH	2AH	12800H to 12BFFH	4AH	1A800H to 1ABFFH	6AH
02C00H to 02FFFH	0BH	0AC00H to 0AFFFH	2BH	12C00H to 12FFFH	4BH	1AC00H to 1AFFFH	6BH
03000H to 033FFH	0CH	0B000H to 0B3FFH	2CH	13000H to 133FFH	4CH	1B000H to 1B3FFH	6CH
03400H to 037FFH	0DH	0B400H to 0B7FFH	2DH	13400H to 137FFH	4DH	1B400H to 1B7FFH	6DH
03800H to 03BFFH	0EH	0B800H to 0BBFFH	2EH	13800H to 13BFFH	4EH	1B800H to 1BBFFH	6EH
03C00H to 03FFFH	0FH	0BC00H to 0BFFFH	2FH	13C00H to 13FFFH	4FH	1BC00H to 1BFFFH	6FH
04000H to 043FFH	10H	0C000H to 0C3FFH	30H	14000H to 143FFH	50H	1C000H to 1C3FFH	70H
04400H to 047FFH	11H	0C400H to 0C7FFH	31H	14400H to 147FFH	51H	1C400H to 1C7FFH	71H
04800H to 04BFFH	12H	0C800H to 0CBFFH	32H	14800H to 14BFFH	52H	1C800H to 1CBFFH	72H
04C00H to 04FFFH	13H	0CC00H to 0CFFFH	33H	14C00H to 14FFFH	53H	1CC00H to 1CFFFH	73H
05000H to 053FFH	14H	0D000H to 0D3FFH	34H	15000H to 153FFH	54H	1D000H to 1D3FFH	74H
05400H to 057FFH	15H	0D400H to 0D7FFH	35H	15400H to 157FFH	55H	1D400H to 1D7FFH	75H
05800H to 05BFFH	16H	0D800H to 0DBFFH	36H	15800H to 15BFFH	56H	1D800H to 1DBFFH	76H
05C00H to 05FFFH	17H	0DC00H to 0DFFFH	37H	15C00H to 15FFFH	57H	1DC00H to 1DFFFH	77H
06000H to 063FFH	18H	0E000H to 0E3FFH	38H	16000H to 163FFH	58H	1E000H to 1E3FFH	78H
06400H to 067FFH	19H	0E400H to 0E7FFH	39H	16400H to 167FFH	59H	1E400H to 1E7FFH	79H
06800H to 06BFFH	1AH	0E800H to 0EBFFH	3AH	16800H to 16BFFH	5AH	1E800H to 1EBFFH	7AH
06C00H to 06FFFH	1BH	0EC00H to 0EFFFH	3BH	16C00H to 16FFFH	5BH	1EC00H to 1EFFFH	7BH
07000H to 073FFH	1CH	0F000H to 0F3FFH	3CH	17000H to 173FFH	5CH	1F000H to 1F3FFH	7CH
07400H to 077FFH	1DH	0F400H to 0F7FFH	3DH	17400H to 177FFH	5DH	1F400H to 1F7FFH	7DH
07800H to 07BFFH	1EH	0F800H to 0FBFFH	3EH	17800H to 17BFFH	5EH	1F800H to 1FBFFH	7EH
07C00H to 07FFFH	1FH	0FC00H to 0FFFFH	3FH	17C00H to 17FFFH	5FH	1FC00H to 1FFFFH	7FH

**Remark** R5F10WLA, R5F10WMA: Block numbers 00H to 0FH  
R5F10WLC, R5F10WMC: Block numbers 00H to 1FH  
R5F10WLD, R5F10WMD: Block numbers 00H to 2FH  
R5F10WLE, R5F10WME: Block numbers 00H to 3FH  
R5F10WLF, R5F10WMF: Block numbers 00H to 5FH  
R5F10WLG, R5F10WMG: Block numbers 00H to 7FH

### 3.1.1 Internal program memory space

The internal program memory space stores the program and table data. The RL78/L13 products incorporate internal ROM (flash memory), as shown below.

**Table 3-2. Internal ROM Capacity**

Part Number	Internal ROM	
	Structure	Capacity
R5F10WLA, R5F10WMA	Flash memory	16384 × 8 bits (00000H to 03FFFFH)
R5F10WLC, R5F10WMC		32768 × 8 bits (00000H to 07FFFFH)
R5F10WLD, R5F10WMD		49152 × 8 bits (00000H to 0BFFFFH)
R5F10WLE, R5F10WME		65536 × 8 bits (00000H to 0FFFFFFH)
R5F10WLF, R5F10WMF		98304 × 8 bits (00000H to 17FFFFH)
R5F10WLG, R5F10WMG		131072 × 8 bits (00000H to 1FFFFFFH)

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 64 K address of 00000H to 0FFFFFFH, because the vector code is assumed to be 2 bytes.

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)

&lt;R&gt;

Vector Table Address	Interrupt Source	80-pin	64-pin
00000H	RESET, POR, LVD, WDT, TRAP, IAW, RPE	√	√
00004H	INTWDTI	√	√
00006H	INTLVI	√	√
00008H	INTP0	√	√
0000AH	INTP1	√	√
0000CH	INTP2	√	√
0000EH	INTP3	√	√
00010H	INTP4	√	√
00012H	INTP5	√	√
00014H	INTST2	√	√
00016H	INTSR2	√	√
00018H	INTSRE2	√	√
0001AH	INTDMA0	√	√
0001CH	INTDMA1	√	√
0001EH	INTST0/INTCSI00/INTIIC00	√	√
00020H	INTTM00	√	√
00022H	INTSR0	√	√
00024H	INTSRE0	√	√
	INTTM01H	√	√
00026H	INTST1/INTCSI10/INTIIC10	√	√
00028H	INTSR1	√	√
0002AH	INTSRE1	√	√
	INTTM03H	√	√
0002CH	INTIICA0	√	√
0002EH	INTRTIT	√	√
00032H	INTTM01	√	√
00034H	INTTM02	√	√
00036H	INTTM03	√	√
00038H	INTAD	√	√
0003AH	INTRTC	√	√
0003CH	INTIT	√	√
0003EH	INTKR	√	√
00040H	INTST3	√	–
00042H	INTSR3	√	–
00044H	INTTKB20	√	√



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

Vector Table Address	Interrupt Source	80-pin	64-pin
00046H	INTTM04	√	√
00048H	INTTM05	√	√
0004AH	INTP6	√	√
0004CH	INTP7	√	√
00050H	INTCMP0	√	√
00052H	INTCMP1	√	√
00054H	INTTM06	√	√
00056H	INTTM07	√	√
0005CH	INTSRE3	√	–
0005EH	INTMD	√	√
00062H	INTFL	√	√
00064H	INTDMA2	√	√
00066H	INTDMA3	√	√
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 0FFFFH (because an address code is of 2 bytes). 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 27 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 29 ON-CHIP DEBUG FUNCTION**.

### 3.1.2 Mirror area

The RL78/L13 mirrors the code flash area of 00000H to 0FFFFH, to F0000H to FFFFFH. The products with 96 KB or more flash memory mirror the code flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, to F0000H to FFFFFH (the code flash area to be mirrored is set by the processor mode control register (PMC)).

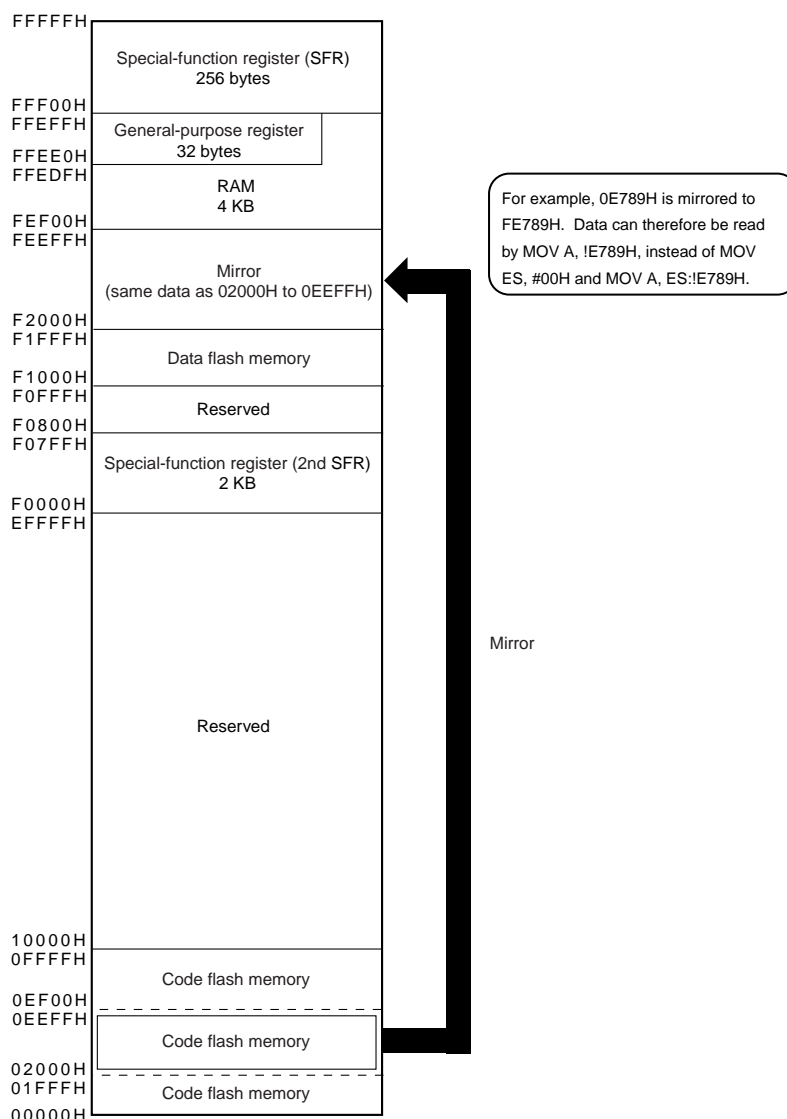
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the SFR, extended SFR, RAM area, data flash area, and use prohibited areas.

See **3.2 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 R5F10WLE, R5F10WME (Flash memory: 64 KB, RAM: 4 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 3-7. 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	10000H to 1FFFFH is mirrored to F0000H to FFFFFH <sup>Note</sup>

**Note** This setting is prohibited in products with 64 KB or less flash memory

- Cautions**
1. In products with 64 KB or less flash memory, be sure to clear bit 0 (MAA) of this register to 0 (default value).
  2. After setting the PMC register, wait for at least one instruction and access the mirror area.

### 3.1.3 Internal data memory space

The RL78/L13 products incorporate the following RAMs.

**Table 3-4. Internal RAM Capacity**

Part Number	Internal RAM
R5F10WLA, R5F10WMA	1024 × 8 bits (FFB00H to FFEFFH)
R5F10WLC, R5F10WMC	1536 × 8 bits (FF900H to FFEFFH)
R5F10WLD, R5F10WMD	2048 × 8 bits (FF700H to FFEFFH)
R5F10WLE, R5F10WME	4096 × 8 bits (FEF00H to FFEFFH)
R5F10WLF, R5F10WMF	6144 × 8 bits (FE700H to FFEFFH)
R5F10WLG, R5F10WVG	8192 × 8 bits (FDF00H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are written and executed. 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. However, instructions cannot be executed by using the general-purpose registers.

The internal RAM is used as stack memory.

- Cautions**
1. The space (FFEE0H to FFEFFH) that the general-purpose registers are allocated cannot be used 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 these areas are used for each library.

R5F10WLG, R5F10WVG: FDF00H to FE309H

### 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.

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

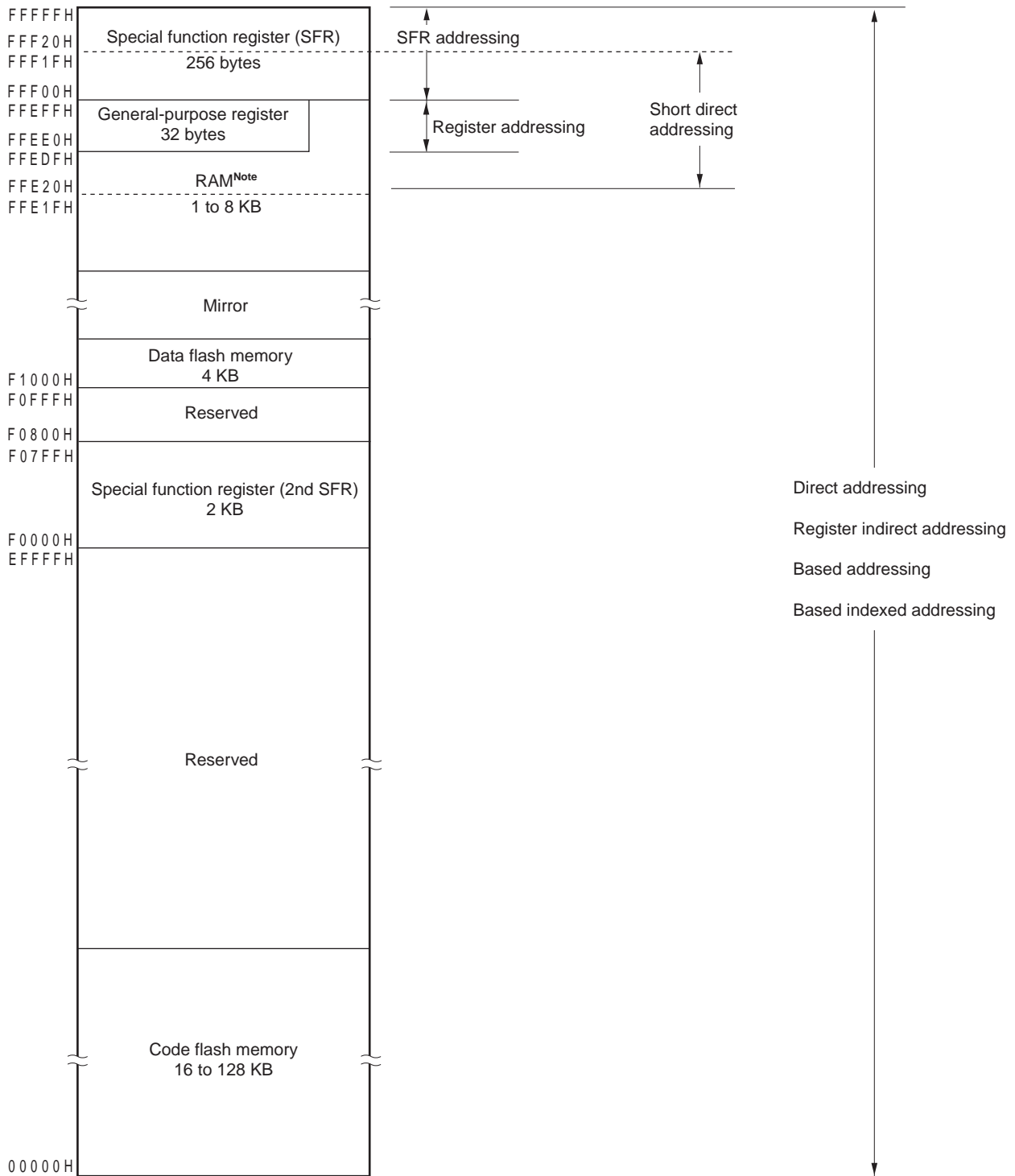
### 3.1.6 Data memory addressing

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

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the RL78/L13, 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 3-8 shows correspondence between data memory and addressing.

For details of each addressing, see **3.5 Addressing for Processing Data Addresses**.

Figure 3-8. Correspondence Between Data Memory and Addressing



### 3.2 Processor Registers

The RL78/L13 products incorporate the following processor registers.

#### 3.2.1 Control registers

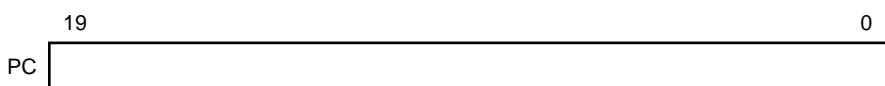
The control registers control the program sequence, statuses and stack memory. The control registers consist of a program counter (PC), a program status word (PSW) and a stack pointer (SP).

##### (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.

<R> Reset signal generation sets the reset vector table values at addresses 00000H and 00001H to the program counter.

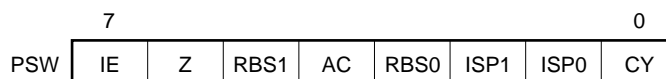
Figure 3-9. Format of Program Counter



##### (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 3-10. Format of Program Status Word



##### (a) Interrupt enable flag (IE)

This flag controls the interrupt request acknowledge operations of the CPU. When 0, the IE flag is set to the interrupt disabled (DI) state, and all maskable interrupt requests are disabled. When 1, the IE flag is set to the interrupt enabled (EI) state and maskable interrupt request acknowledgment is controlled with an in-service priority flag (ISP1, ISPO), 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 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 **19.3.3**) can not be acknowledged. Actual vectored interrupt request acknowledgment is controlled by the interrupt enable flag (IE).

**Remark** n = 0, 1

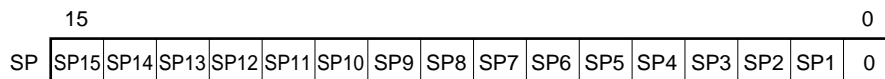
**(f) Carry flag (CY)**

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 3-11. Format of Stack Pointer**



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 areas of the following products is prohibited when performing self-programming and rewriting the data flash memory, because these areas are used for each library.

R5F10WLG, R5F10WVG: FDF00H to FE309H



### 3.2.2 General-purpose registers

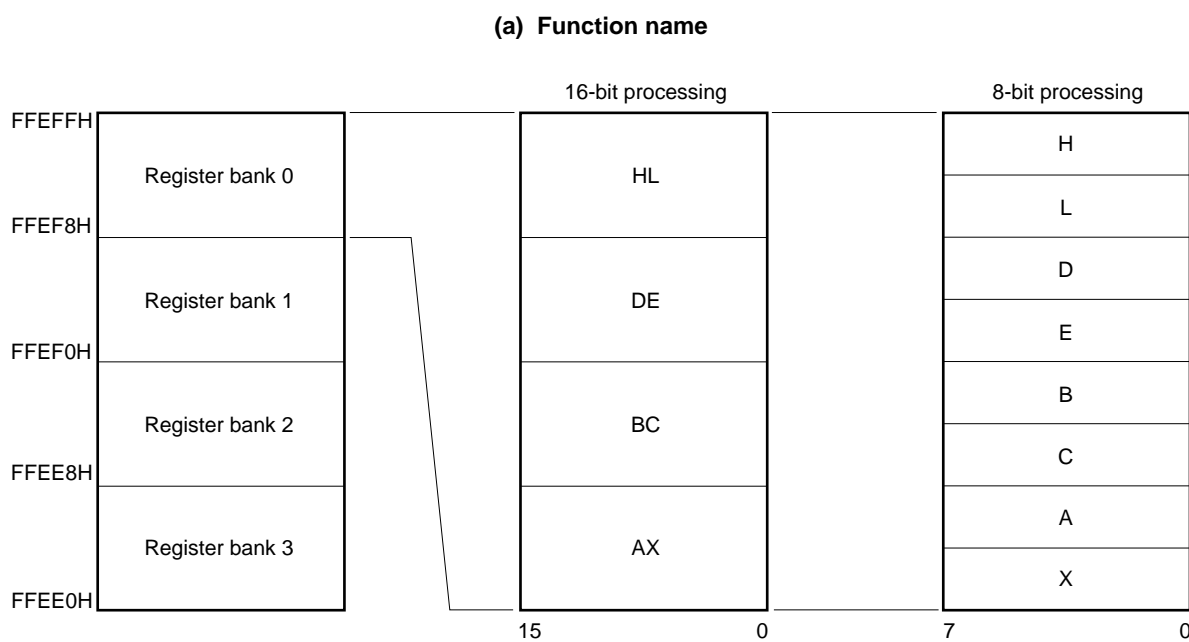
General-purpose registers are mapped at particular addresses (FFEE0H to FFEFFH) of the data memory. The general-purpose registers consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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 3-12. Configuration of General-Purpose Registers

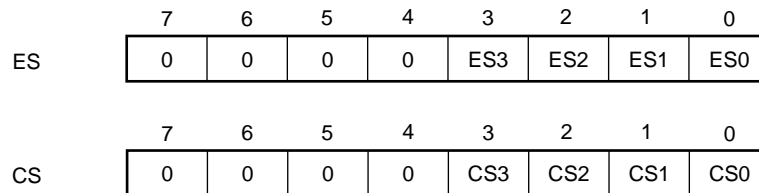


### 3.2.3 ES and CS registers

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

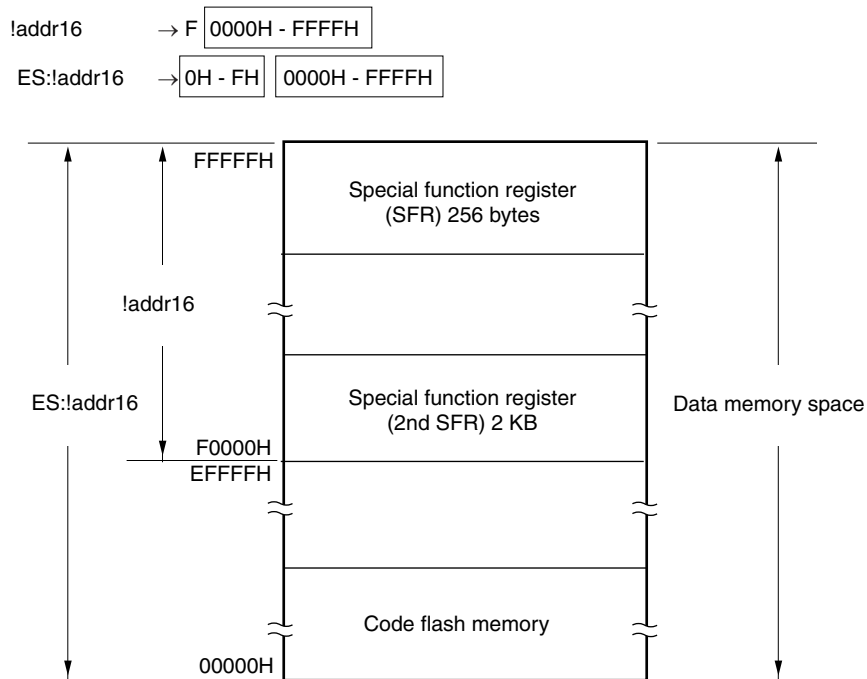
The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

**Figure 3-13. Configuration of ES and CS Registers**



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

**Figure 3-14. 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 reserved 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 reserved 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 extended SFRs are not assigned.

**Remark** For extended SFRs (2nd SFRs), see 3.3.5 **Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 3-5. SFR List (1/5)

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
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	–	√	√	0000H
FFF11H		–			–	–		
FFF12H	Serial data register 01	RXD0	SDR01	R/W	–	√	√	0000H
FFF13H		–			–	–		
FFF14H	Serial data register 12	TXD3	SDR12	R/W	–	√	√	0000H
FFF15H		–			–	–		
FFF16H	Serial data register 13	RXD3	SDR13	R/W	–	√	√	0000H
FFF17H		–			–	–		
FFF18H	Timer data register 00	TDR00		R/W	–	–	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	–	√	√	00H
FFF1BH		TDR01H			–	√	–	
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
FFF2DH	Port mode register 13	PM13		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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF34H	Key return control register	KRCTL		R/W	√	√	–	00H
FFF35H	Key return flag register	KRF		R/W	√	√	–	00H
FFF37H	Key return mode register	KRM0		R/W	√	√	–	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	–	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	–	00H
FFF40H	LCD mode register 0	LCDM0		R/W	–	√	–	00H
FFF41H	LCD mode register 1	LCDM1		R/W	√	√	–	00H
FFF42H	LCD clock control register	LCDC0		R/W	–	√	–	00H
FFF43H	LCD boost level control register	VLCD		R/W	–	√	–	04H
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	–	√	√	0000H
FFF45H		–			–	–		
FFF46H	Serial data register 03	RXD1	SDR03	R/W	–	√	√	0000H
FFF47H		–			–	–		
FFF48H	Serial data register 10	TXD2	SDR10	R/W	–	√	√	0000H
FFF49H		–			–	–		
FFF4AH	Serial data register 11	RXD2	SDR11	R/W	–	√	√	0000H
FFF4BH		–			–	–		
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
FFF64H	Timer data register 02	TDR02		R/W	–	–	√	0000H
FFF65H					–	–	–	
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	–	√	√	00H
FFF67H		TDR03H			–	√	00H	
FFF68H	Timer data register 04	TDR04		R/W	–	–	√	0000H
FFF69H					–	–	–	
FFF6AH	Timer data register 05	TDR05		R/W	–	–	√	0000H
FFF6BH					–	–	–	
FFF6CH	Timer data register 06	TDR06		R/W	–	–	√	0000H
FFF6DH					–	–	–	
FFF6EH	Timer data register 07	TDR07		R/W	–	–	√	0000H
FFF6FH					–	–	–	

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
FFF90H FFF91H	12-bit interval timer control register	ITMC	R/W	–	–	√	0FFFH
FFF92H	Second count register	SEC	R/W	–	√	–	Undefined
FFF93H	Minute count register	MIN	R/W	–	√	–	Undefined
FFF94H	Hour count register	HOUR	R/W	–	√	–	Undefined
FFF95H	Week count register	WEEK	R/W	–	√	–	Undefined
FFF96H	Day count register	DAY	R/W	–	√	–	Undefined
FFF97H	Month count register	MONTH	R/W	–	√	–	Undefined
FFF98H	Year count register	YEAR	R/W	–	√	–	Undefined
FFF9AH	Alarm minute register	ALARMWM	R/W	–	√	–	Undefined
FFF9BH	Alarm hour register	ALARMWH	R/W	–	√	–	Undefined
FFF9CH	Alarm week register	ALARMWW	R/W	–	√	–	Undefined
FFF9DH	Real-time clock control register 0	RTCC0	R/W	√	√	–	00H <sup>Note 1</sup>
FFF9EH	Real-time clock control register 1	RTCC1	R/W	√	√	–	00H <sup>Note 1</sup>
FFFA0H	Clock operation mode control register	CMC	R/W	–	√	–	00H <sup>Note 1</sup>
FFFA1H	Clock operation status control register	CSC	R/W	√	√	–	C0H <sup>Note 1</sup>
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	–	√	–	Undefined <sup>Note 2</sup>
FFFA9H	Voltage detection register	LVIM	R/W	√	√	–	00H <sup>Note 2</sup>
FFFAAH	Voltage detection level register	LVIS	R/W	√	√	–	00H/01H/81H <sup>Note 2</sup>

- Notes**
1. This register is reset only by a power-on reset.
  2. The reset values of the registers vary depending on the reset source as shown below.

Reset Source		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
Register	RESF	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				
LVIM	LVISEN bit	Cleared (0)						Held
	LVIOMSK bit			Held				
	LVIF bit							
LVIS		Cleared (00H/01H/81H)						

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFABH	Watchdog timer enable register	WDTE		R/W	–	√	–	1AH/9AH <sup>Note</sup>
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 0L	DRA0L	DRA0	R/W	–	√	√	00H
FFFB3H	DMA RAM address register 0H	DRA0H		R/W	–	√		00H
FFFB4H	DMA RAM address register 1L	DRA1L	DRA1	R/W	–	√	√	00H
FFFB5H	DMA RAM address register 1H	DRA1H		R/W	–	√		00H
FFFB6H	DMA byte count register 0L	DBC0L	DBC0	R/W	–	√	√	00H
FFFB7H	DMA byte count register 0H	DBC0H		R/W	–	√		00H
FFFB8H	DMA byte count register 1L	DBC1L	DBC1	R/W	–	√	√	00H
FFFB9H	DMA byte count register 1H	DBC1H		R/W	–	√		00H
FFFBAAH	DMA mode control register 0	DMC0		R/W	√	√	–	00H
FFFBABH	DMA mode control register 1	DMC1		R/W	√	√	–	00H
FFFBCH	DMA operation control register 0	DRC0		R/W	√	√	–	00H
FFFBDAH	DMA operation control register 1	DRC1		R/W	√	√	–	00H
FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD1H	Interrupt request flag register 2H	IF2H		R/W	√	√		00H
FFFD2H	Interrupt request flag register 3L	IF3L	IF3	R/W	√	√	√	00H
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	Interrupt mask flag register 2H	MK2H		R/W	√	√		FFH
FFFD6H	Interrupt mask flag register 3L	MK3L	MK3	R/W	√	√	√	FFH
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	Priority specification flag register 02H	PR02H		R/W	√	√		FFH
FFFDAH	Priority specification flag register 03L	PR03L	PR03	R/W	√	√	√	FFH
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
FFDDH	Priority specification flag register 12H	PR12H		R/W	√	√		FFH
FFFDEH	Priority specification flag register 13L	PR13L	PR13	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

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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
FFFE BH	Priority specification flag register 01H	PR01H		R/W	√	√		FFH
FFFECH	Priority specification flag register 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFE DH	Priority specification flag register 10H	PR10H		R/W	√	√		FFH
FFFE EH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFE FH	Priority specification flag register 11H	PR11H		R/W	√	√		FFH
FFFF0H	Multiplication/division data register A (L)	MDAL		R/W	-	-	√	0000H
FFFF1H								
FFFF2H	Multiplication/division data register A (H)	MDAH		R/W	-	-	√	0000H
FFFF3H								
FFFF4H	Multiplication/division data register B (H)	MDBH		R/W	-	-	√	0000H
FFFF5H								
FFFF6H	Multiplication/division data register B (L)	MDBL		R/W	-	-	√	0000H
FFFF7H								
FFFEH	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 (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 reserved 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 reserved 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.

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

**Remark** For SFRs in the SFR area, see 3.3.4 Special function registers (SFRs).

Table 3-6. Extended SFR (2nd SFR) List (1/9)

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
F0030H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
F0032H	Pull-up resistor option register 2	PU2	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
F003DH	Pull-up resistor option register 13	PU13	R/W	√	√	–	00H
F0040H	Port input mode register 0	PIM0	R/W	√	√	–	00H
F0041H	Port input mode register 1	PIM1	R/W	√	√	–	00H
F0043H	Port input mode register 3	PIM3	R/W	√	√	–	00H
F0044H	Port input mode register 4	PIM4	R/W	√	√	–	00H
F0045H	Port input mode register 5	PIM5	R/W	√	√	–	00H
F0050H	Port output mode register 0	POM0	R/W	√	√	–	00H
F0051H	Port output mode register 1	POM1	R/W	√	√	–	00H
F0053H	Port output mode register 3	POM3	R/W	√	√	–	00H
F0054H	Port output mode register 4	POM4	R/W	√	√	–	00H
F0055H	Port output mode register 5	POM5	R/W	√	√	–	00H
F005DH	Port output mode register 13	POM13	R/W	√	√	–	00H
F0061H	Port mode control register 1	PMC1	R/W	√	√	–	FFH
F0062H	Port mode control register 2	PMC2	R/W	√	√	–	FFH
F0064H	Port mode control register 4	PMC4	R/W	√	√	–	00H
F0070H	Noise filter enable register 0	NFEN0	R/W	√	√	–	00H
F0071H	Noise filter enable register 1	NFEN1	R/W	√	√	–	00H
F0073H	Input switch control register	ISC	R/W	√	√	–	00H
F0074H	Timer input select register 0	TIS0	R/W	–	√	–	00H
F0076H	A/D port configuration register	ADPC	R/W	–	√	–	00H
F0077H	Peripheral I/O redirection register	PIOR	R/W	–	√	–	00H
F0078H	Invalid memory access detection control register	IAWCTL	R/W	–	√	–	00H
F0079H	Timer output select register	TOS	R/W	√	√	–	00H
F007AH	Peripheral enable register 1	PER1	R/W	√	√	–	00H
F007BH	Port mode select register	PMS	R/W	√	√	–	00H

Table 3-6. Extended SFR (2nd SFR) List (2/9)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	
				1-bit	8-bit	16-bit		
F007CH	Timer KB2 output pin switch register	PWCTKB	R/W	–	√	–	00H	
F0090H	Data flash control register	DFLCTL	R/W	√	√	–	00H	
F00A0H	High-speed on-chip oscillator trimming register	HIOTRM	R/W	–	√	–	Undefined <sup>Note 1</sup>	
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV	R/W	–	√	–	Undefined <sup>Note 2</sup>	
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	Subsystem clock supply mode control register	OSMC	R/W	–	√	–	00H	
F00F5H	RAM parity error control register	RPECTL	R/W	√	√	–	00H	
F00F9H	Power-on-reset status register	PORSR	R/W	–	√	–	00H <sup>Note 3</sup>	
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		–			–			
F0104H	Serial status register 02	SSR02L	SSR02	R	–	√	√	0000H
F0105H		–			–			
F0106H	Serial status register 03	SSR03L	SSR03	R	–	√	√	0000H
F0107H		–			–			
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		–			–			
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	–	√	√	0000H
F010DH		–			–			
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	–	√	√	0000H
F010FH		–			–			
F0110H	Serial mode register 00	SMR00	R/W	–	–	√	0020H	
F0111H								
F0112H	Serial mode register 01	SMR01	R/W	–	–	√	0020H	
F0113H								
F0114H	Serial mode register 02	SMR02	R/W	–	–	√	0020H	
F0115H								
F0116H	Serial mode register 03	SMR03	R/W	–	–	√	0020H	
F0117H								

**Notes** 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).

3. This register is reset only by a power-on reset.

Table 3-6. Extended SFR (2nd SFR) List (3/9)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	–	–	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	–	–	√	0087H
F011FH								
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		–			–			
F0128H	Serial output register 0	SO0		R/W	–	–	√	0F0FH
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
F0139H		–			–			
F0140H	Serial status register 10	SSR10L	SSR10	R	–	√	√	0000H
F0141H		–			–			
F0142H	Serial status register 11	SSR11L	SSR11	R	–	√	√	0000H
F0143H		–			–			
F0144H	Serial status register 12	SSR12L	SSR12	R	–	√	√	0000H
F0145H		–			–			
F0146H	Serial status register 13	SSR13L	SSR13	R	–	√	√	0000H
F0147H		–			–			
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	–	√	√	0000H
F0149H		–			–			
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	–	√	√	0000H
F014BH		–			–			
F014CH	Serial flag clear trigger register 12	SIR12L	SIR12	R/W	–	√	√	0000H
F014DH		–			–			
F014EH	Serial flag clear trigger register 13	SIR13L	SIR13	R/W	–	√	√	0000H
F014FH		–			–			
F0150H	Serial mode register 10	SMR10		R/W	–	–	√	0020H
F0151H								

Table 3-6. Extended SFR (2nd SFR) List (4/9)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0152H	Serial mode register 11	SMR11		R/W	–	–	√	0020H
F0153H								
F0154H	Serial mode register 12	SMR12		R/W	–	–	√	0020H
F0155H								
F0156H	Serial mode register 13	SMR13		R/W	–	–	√	0020H
F0157H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	–	–	√	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	–	–	√	0087H
F015BH								
F015CH	Serial communication operation setting register 12	SCR12		R/W	–	–	√	0087H
F015DH								
F015EH	Serial communication operation setting register 13	SCR13		R/W	–	–	√	0087H
F015FH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	√	√	√	0000H
F0161H		–			–	–		
F0162H	Serial channel start register 1	SS1L	SS1	R/W	√	√	√	0000H
F0163H		–			–	–		
F0164H	Serial channel stop register 1	ST1L	ST1	R/W	√	√	√	0000H
F0165H		–			–	–		
F0166H	Serial clock select register 1	SPS1L	SPS1	R/W	–	√	√	0000H
F0167H		–			–	–		
F0168H	Serial output register 1	SO1		R/W	–	–	√	0F0FH
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1	R/W	√	√	√	0000H
F016BH		–			–	–		
F0174H	Serial output level register 1	SOL1L	SOL1	R/W	–	√	√	0000H
F0175H		–			–	–		
F0178H	Serial standby control register 1	SSC1L	SSC1	R/W	–	√	√	0000H
F0179H		–			–	–		
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								
F0188H	Timer counter register 04	TCR04		R	–	–	√	FFFFH
F0189H								

Table 3-6. Extended SFR (2nd SFR) List (5/9)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F018AH	Timer counter register 05	TCR05		R	–	–	√	FFFFH
F018BH								
F018CH	Timer counter register 06	TCR06		R	–	–	√	FFFFH
F018DH								
F018EH	Timer counter register 07	TCR07		R	–	–	√	FFFFH
F018FH								
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								
F0198H	Timer mode register 04	TMR04		R/W	–	–	√	0000H
F0199H								
F019AH	Timer mode register 05	TMR05		R/W	–	–	√	0000H
F019BH								
F019CH	Timer mode register 06	TMR06		R/W	–	–	√	0000H
F019DH								
F019EH	Timer mode register 07	TMR07		R/W	–	–	√	0000H
F019FH								
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		–			–			
F01A8H	Timer status register 04	TSR04L	TSR04	R	–	√	√	0000H
F01A9H		–			–			
F01AAH	Timer status register 05	TSR05L	TSR05	R	–	√	√	0000H
F01ABH		–			–			
F01ACH	Timer status register 06	TSR06L	TSR06	R	–	√	√	0000H
F01ADH		–			–			
F01AEH	Timer status register 07	TSR07L	TSR07	R	–	√	√	0000H
F01AFH		–			–			
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H
F01B1H		–			–			
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	√	√	√	0000H
F01B3H		–			–			

Table 3-6. Extended SFR (2nd SFR) List (6/9)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		–			–			
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		–			–			
F0200H	DMA SFR address register 2	DSA2		R/W	–	√	–	00H
F0201H	DMA SFR address register 3	DSA3		R/W	–	√	–	00H
F0202H	DMA RAM address register 2L	DRA2L	DRA2	R/W	–	√	√	00H
F0203H	DMA RAM address register 2H	DRA2H		R/W	–	√		
F0204H	DMA RAM address register 3L	DRA3L	DRA3	R/W	–	√	√	00H
F0205H	DMA RAM address register 3H	DRA3H		R/W	–	√		
F0206H	DMA byte count register 2L	DBC2L	DBC2	R/W	–	√	√	00H
F0207H	DMA byte count register 2H	DBC2H		R/W	–	√		
F0208H	DMA byte count register 3L	DBC3L	DBC3	R/W	–	√	√	00H
F0209H	DMA byte count register 3H	DBC3H		R/W	–	√		
F020AH	DMA mode control register 2	DMC2		R/W	√	√	–	00H
F020BH	DMA mode control register 3	DMC3		R/W	√	√	–	00H
F020CH	DMA operation control register 2	DRC2		R/W	√	√	–	00H
F020DH	DMA operation control register 3	DRC3		R/W	√	√	–	00H
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
F0240H	TMKB2 counter restart trigger select register 00	ELSELR00		R/W	–	√	–	00H
F0241H	TMKB2 counter restart trigger select register 01	ELSELR01		R/W	–	√	–	00H
F0242H	TMKB2 counter restart trigger select register 02	ELSELR02		R/W	–	√	–	00H
F0243H	TMKB2 counter restart trigger select register 03	ELSELR03		R/W	–	√	–	00H
F0244H	TMKB2 counter restart trigger select register 04	ELSELR04		R/W	–	√	–	00H

Table 3-6. Extended SFR (2nd SFR) List (7/9)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0245H	TMKB2 counter restart trigger select register 05	ELSELR05	R/W	–	√	–	00H
F0246H	TMKB2 counter restart trigger select register 06	ELSELR06	R/W	–	√	–	00H
F0247H	TMKB2 counter restart trigger select register 07	ELSELR07	R/W	–	√	–	00H
F0248H	TMKB2 counter restart trigger select register 08	ELSELR08	R/W	–	√	–	00H
F0249H	TMKB2 counter restart trigger select register 09	ELSELR09	R/W	–	√	–	00H
F02F0H	Flash memory CRC control register	CRC0CTL	R/W	√	√	–	00H
F02F2H	Flash memory CRC operation result register	PGCRCL	R/W	–	–	√	0000H
F02FAH	CRC data register	CRCD	R/W	–	–	√	0000H
F0300H	LCD port function register 0	PFSEG0	R/W	√	√	–	F0H
F0301H	LCD port function register 1	PFSEG1	R/W	√	√	–	FFH
F0302H	LCD port function register 2	PFSEG2	R/W	√	√	–	FFH
F0303H	LCD port function register 3	PFSEG3	R/W	√	√	–	FFH
F0304H	LCD port function register 4	PFSEG4	R/W	√	√	–	FFH
F0305H	LCD port function register 5	PFSEG5	R/W	√	√	–	FFH
F0306H	LCD port function register 6	PFSEG6	R/W	√	√	–	0FH
F0308H	LCD input switch control register	ISCLCD	R/W	√	√	–	00H
F0310H	Watch error correction register	SUBCUD	R/W	–	–	√	0020H <sup>Note</sup>
F0311H							
F0340H	Comparator mode setting register	COMPMDR	R/W	√	√	–	00H
F0341H	Comparator filter control register	COMPFIR	R/W	√	√	–	00H
F0342H	Comparator output control register	COMPOCR	R/W	√	√	–	00H
F0400H	LCD display data memory 0	SEG0	R/W	–	√	–	00H
F0401H	LCD display data memory 1	SEG1	R/W	–	√	–	00H
F0402H	LCD display data memory 2	SEG2	R/W	–	√	–	00H
F0403H	LCD display data memory 3	SEG3	R/W	–	√	–	00H
F0404H	LCD display data memory 4	SEG4	R/W	–	√	–	00H
F0405H	LCD display data memory 5	SEG5	R/W	–	√	–	00H
F0406H	LCD display data memory 6	SEG6	R/W	–	√	–	00H
F0407H	LCD display data memory 7	SEG7	R/W	–	√	–	00H
F0408H	LCD display data memory 8	SEG8	R/W	–	√	–	00H
F0409H	LCD display data memory 9	SEG9	R/W	–	√	–	00H
F040AH	LCD display data memory 10	SEG10	R/W	–	√	–	00H
F040BH	LCD display data memory 11	SEG11	R/W	–	√	–	00H
F040CH	LCD display data memory 12	SEG12	R/W	–	√	–	00H
F040DH	LCD display data memory 13	SEG13	R/W	–	√	–	00H

**Note** This register is reset only by a power-on reset.



Table 3-6. Extended SFR (2nd SFR) List (8/9)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F040EH	LCD display data memory 14	SEG14	R/W	–	√	–	00H
F040FH	LCD display data memory 15	SEG15	R/W	–	√	–	00H
F0410H	LCD display data memory 16	SEG16	R/W	–	√	–	00H
F0411H	LCD display data memory 17	SEG17	R/W	–	√	–	00H
F0412H	LCD display data memory 18	SEG18	R/W	–	√	–	00H
F0413H	LCD display data memory 19	SEG19	R/W	–	√	–	00H
F0414H	LCD display data memory 20	SEG20	R/W	–	√	–	00H
F0415H	LCD display data memory 21	SEG21	R/W	–	√	–	00H
F0416H	LCD display data memory 22	SEG22	R/W	–	√	–	00H
F0417H	LCD display data memory 23	SEG23	R/W	–	√	–	00H
F0418H	LCD display data memory 24	SEG24	R/W	–	√	–	00H
F0419H	LCD display data memory 25	SEG25	R/W	–	√	–	00H
F041AH	Use prohibited						
F041BH	LCD display data memory 26	SEG26	R/W	–	√	–	00H
F041CH	LCD display data memory 27	SEG27	R/W	–	√	–	00H
F041DH	LCD display data memory 28	SEG28	R/W	–	√	–	00H
F041EH	LCD display data memory 29	SEG29	R/W	–	√	–	00H
F041FH	LCD display data memory 30	SEG30	R/W	–	√	–	00H
F0420H	LCD display data memory 31	SEG31	R/W	–	√	–	00H
F0421H	LCD display data memory 32	SEG32	R/W	–	√	–	00H
F0422H	LCD display data memory 33	SEG33	R/W	–	√	–	00H
F0423H	LCD display data memory 34	SEG34	R/W	–	√	–	00H
F0424H	LCD display data memory 35	SEG35	R/W	–	√	–	00H
F0425H	LCD display data memory 36	SEG36	R/W	–	√	–	00H
F0426H	LCD display data memory 37	SEG37	R/W	–	√	–	00H
F0427H	LCD display data memory 38	SEG38	R/W	–	√	–	00H
F0428H	LCD display data memory 39	SEG39	R/W	–	√	–	00H
F0429H	LCD display data memory 40	SEG40	R/W	–	√	–	00H
F042AH	LCD display data memory 41	SEG41	R/W	–	√	–	00H
F042BH	LCD display data memory 42	SEG42	R/W	–	√	–	00H
F042CH	LCD display data memory 43	SEG43	R/W	–	√	–	00H
F042DH	LCD display data memory 44	SEG44	R/W	–	√	–	00H
F042EH	LCD display data memory 45	SEG45	R/W	–	√	–	00H
F042FH	LCD display data memory 46	SEG46	R/W	–	√	–	00H
F0430H	LCD display data memory 47	SEG47	R/W	–	√	–	00H
F0431H	LCD display data memory 48	SEG48	R/W	–	√	–	00H
F0432H	LCD display data memory 49	SEG49	R/W	–	√	–	00H
F0433H	LCD display data memory 50	SEG50	R/W	–	√	–	00H
F0500H	16-bit timer KB2 compare register 00	TKBCR00	R/W	–	–	√	0000H
F0501H							
F0502H	16-bit timer KB2 compare register 01	TKBCR01	R/W	–	–	√	0000H
F0503H							
F0504H	16-bit timer KB2 compare register 02	TKBCR02	R/W	–	–	√	0000H
F0505H							

Table 3-6. Extended SFR (2nd SFR) List (9/9)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0506H	16-bit timer KB2 compare register 03	TKBCR03	R/W	–	–	√	0000H
F0507H							
F050AH	16-bit timer KB2 smooth start initial duty register 00	TKBSIR00	R/W	–	–	√	0000H
F050BH							
F050CH	16-bit timer KB2 smooth start initial duty register 01	TKBSIR01	R/W	–	–	√	0000H
F050DH							
F050EH	16-bit timer KB2 dithering count register 00	TKBDNR00	R/W	–	√	–	00H
F050FH	16-bit timer KB2 smooth start step width register 00	TKBSSR00	R/W	–	√	–	00H
F0510H	16-bit timer KB2 dithering count register 01	TKBDNR01	R/W	–	√	–	00H
F0511H	16-bit timer KB2 smooth start step width register 01	TKBSSR01	R/W	–	√	–	00H
F0512H	16-bit timer KB2 trigger register 0	TKBTRG0	R/W	√	√	–	00H
F0513H	16-bit timer KB2 flag register 0	TKBFLG0	R	√	√	–	00H
F0514H	16-bit timer KB2 compare 1L & dithering count register 00	TKBCRLD00	R/W	–	–	√	0000H
F0515H							
F0516H	16-bit timer KB2 compare 1L & dithering count register 01	TKBCRLD01	R/W	–	–	√	0000H
F0517H							
F0520H	16-bit timer counter KB20	TKBCNT0	R	–	–	√	FFFFH
F0522H	16-bit timer KB2 operation control register 00	TKBCTL00	R/W	–	–	√	0000H
F0523H							
F0524H	16-bit timer KB2 maximum frequency limit setting register 0	TKBMFR0	R/W	–	–	√	0000H
F0525H							
F0526H	16-bit timer KB2 output control register 00	TKBIOC00	R/W	√	√	–	00H
F0527H	16-bit timer KB2 flag clear trigger register 0	TKBCLR0	R/W	√	√	–	00H
F0528H	16-bit timer KB2 output control register 01	TKBIOC01	R/W	√	√	–	00H
F0529H	16-bit timer KB2 operation control register 01	TKBCTL01	R/W	√	√	–	00H
F052AH	16-bit timer KB2 count clock division ratio select register 0	TKBPSCS0	R/W	–	√	–	00H
F0530H	Forced output stop function control register 00	TKBPACTL00	R/W	–	–	√	0000H
F0531H							
F0532H	Forced output stop function control register 01	TKBPACTL01	R/W	–	–	√	0000H
F0533H							
F0534H	Forced output stop function 1 start trigger register 0	TKBPAHFS0	R/W	√	√	–	00H
F0535H	Forced output stop function stop trigger register 0	TKBPAHFT0	R/W	√	√	–	00H
F0536H	Forced output stop function flag register 0	TKBPAFLG0	R	√	√	–	00H
F0537H	Forced output stop function control register 02	TKBPACTL02	R/W	√	√	–	00H

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

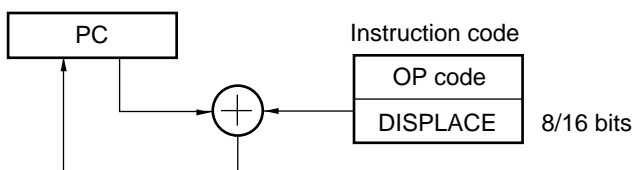
### 3.3 Instruction Address Addressing

#### 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 3-15. Outline of Relative Addressing**



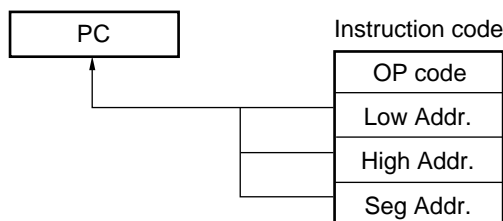
#### 3.3.2 Immediate addressing

**[Function]**

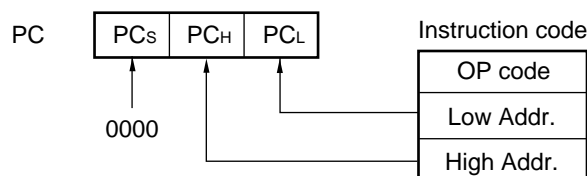
Immediate addressing stores immediate data of the instruction word in the program counter, and specifies the program address to be used as the branch destination.

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 3-16. Example of CALL !!addr20/BR !!addr20**



**Figure 3-17. Example of CALL !addr16/BR !addr16**



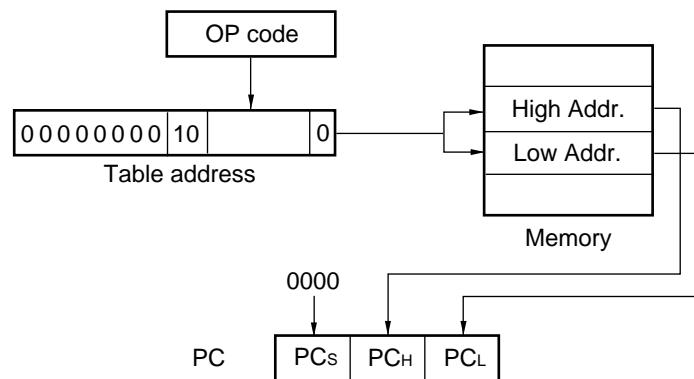
### 3.3.3 Table indirect addressing

#### [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 3-18. Outline of Table Indirect Addressing**

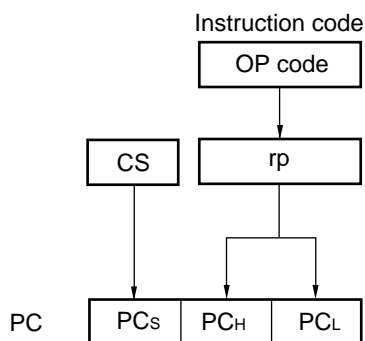


### 3.3.4 Register direct addressing

**[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 3-19. Outline of Register Direct Addressing**



### 3.4 Addressing for Processing Data Addresses

#### 3.4.1 Implied addressing

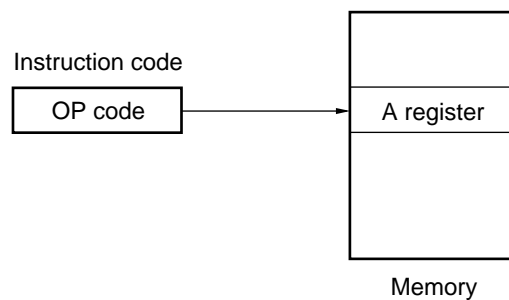
**[Function]**

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 3-20. Outline of Implied Addressing**



#### 3.4.2 Register addressing

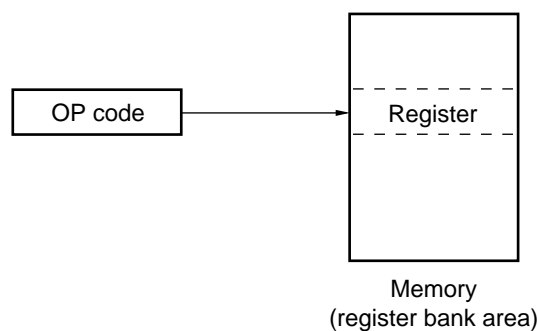
**[Function]**

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 3-21. Outline of Register Addressing**



### 3.4.3 Direct addressing

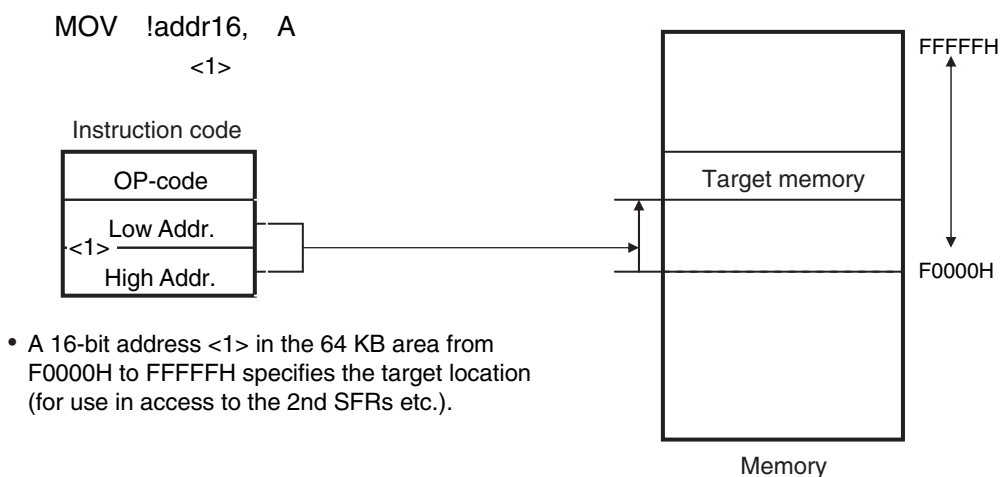
**[Function]**

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

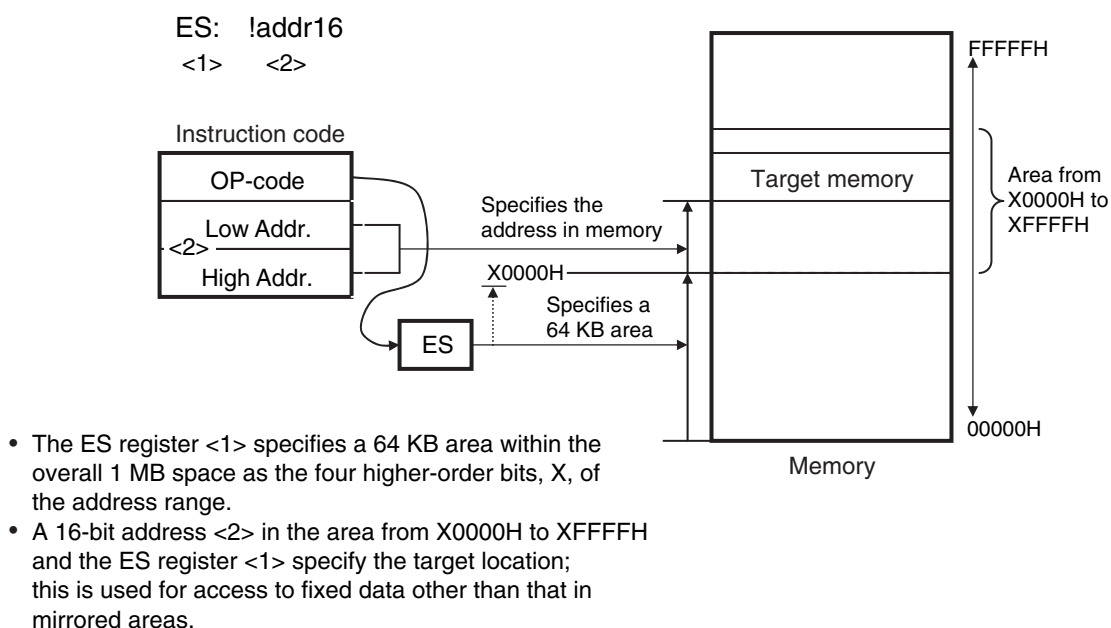
**[Operand format]**

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 3-22. Example of !addr16**



**Figure 3-23. Example of ES:!addr16**



3.4.4 Short direct addressing

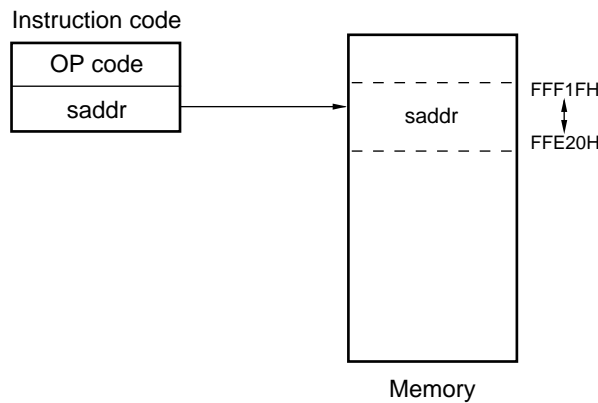
[Function]

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

Figure 3-24. 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**

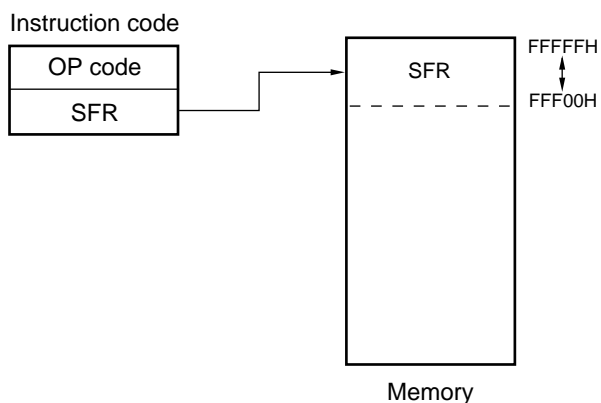
**[Function]**

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)

**Figure 3-25. Outline of SFR Addressing**



### 3.4.6 Register indirect addressing

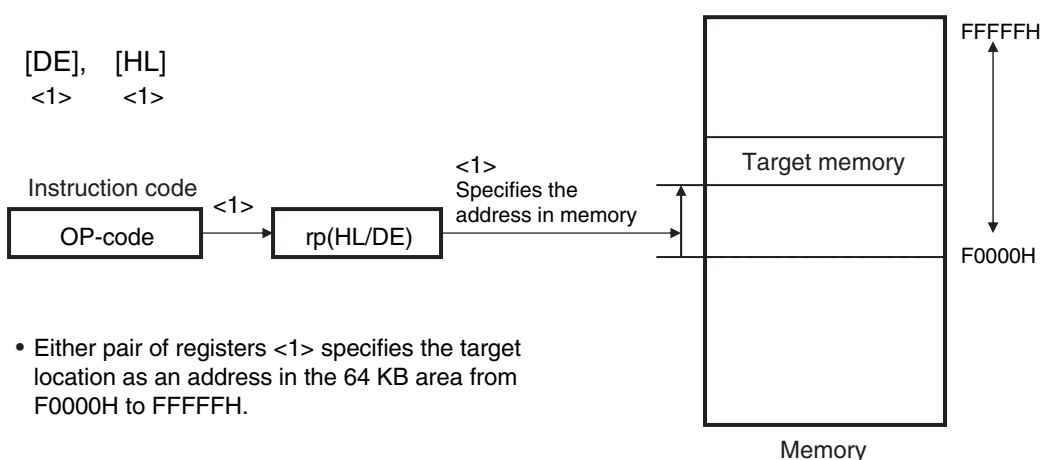
**[Function]**

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

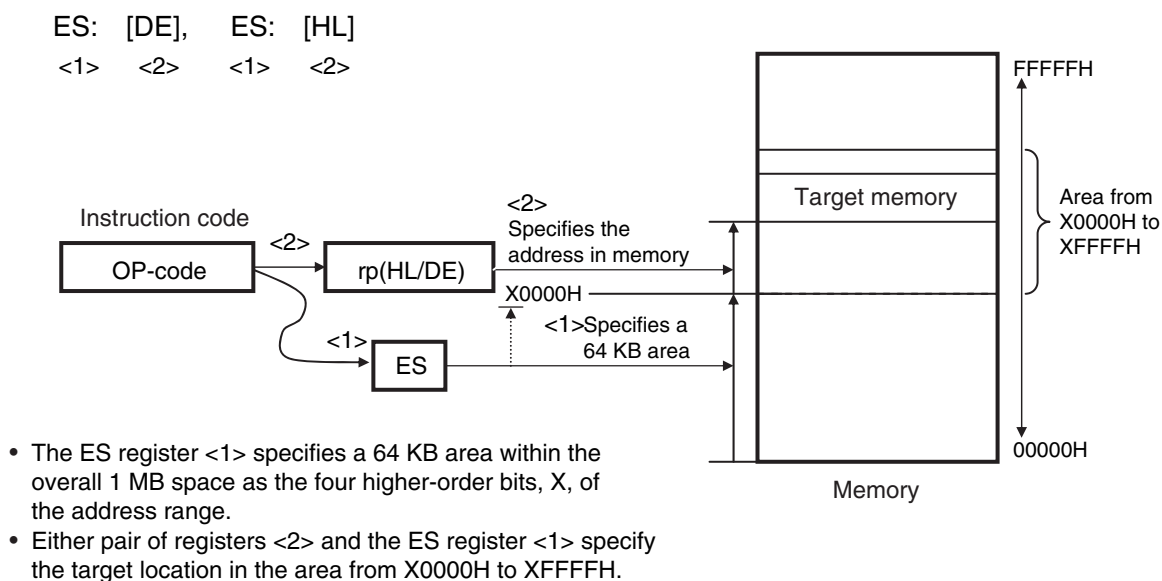
**[Operand format]**

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 3-26. Example of [DE], [HL]**



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



### 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 3-28. Example of [SP + byte]**

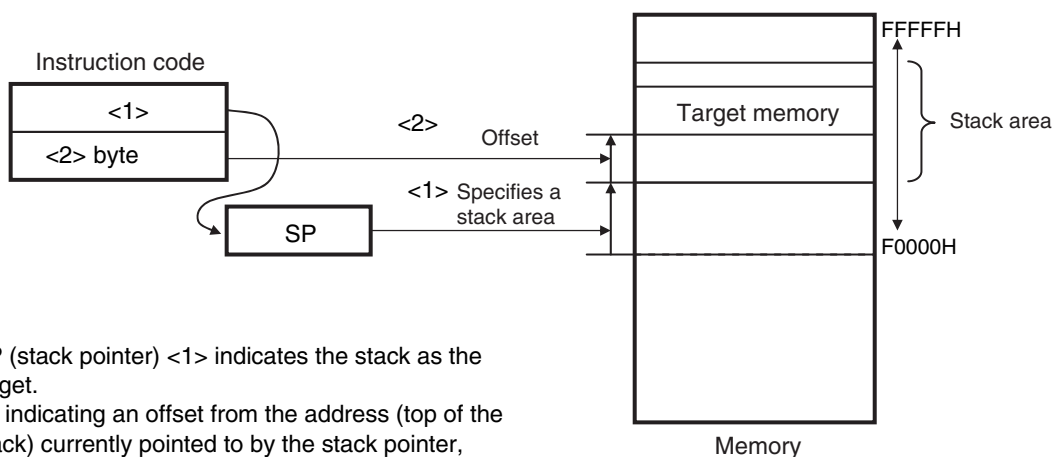


Figure 3-29. Example of [HL + byte], [DE + byte]

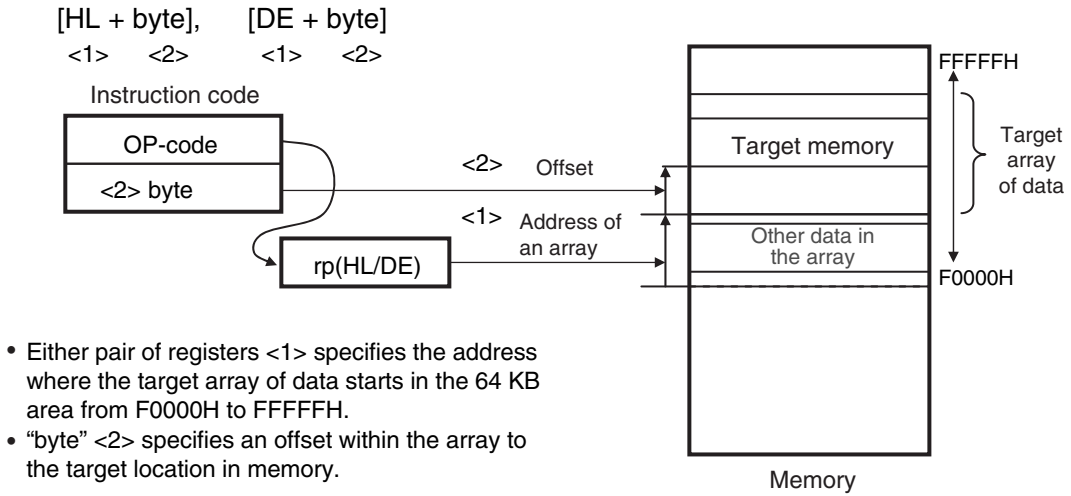


Figure 3-30. Example of word[B], word[C]

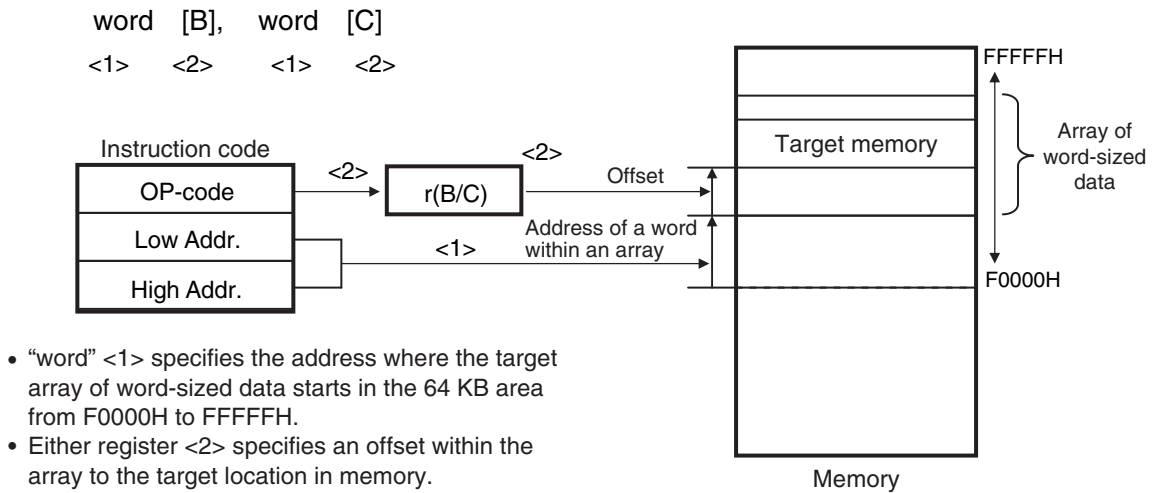


Figure 3-31. Example of word[BC]

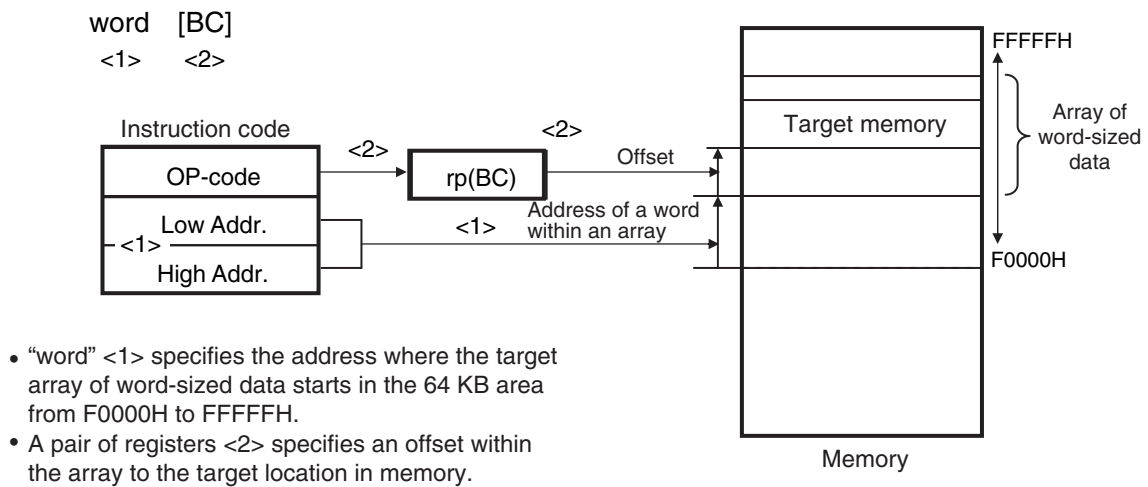


Figure 3-32. Example of ES:[HL + byte], ES:[DE + byte]

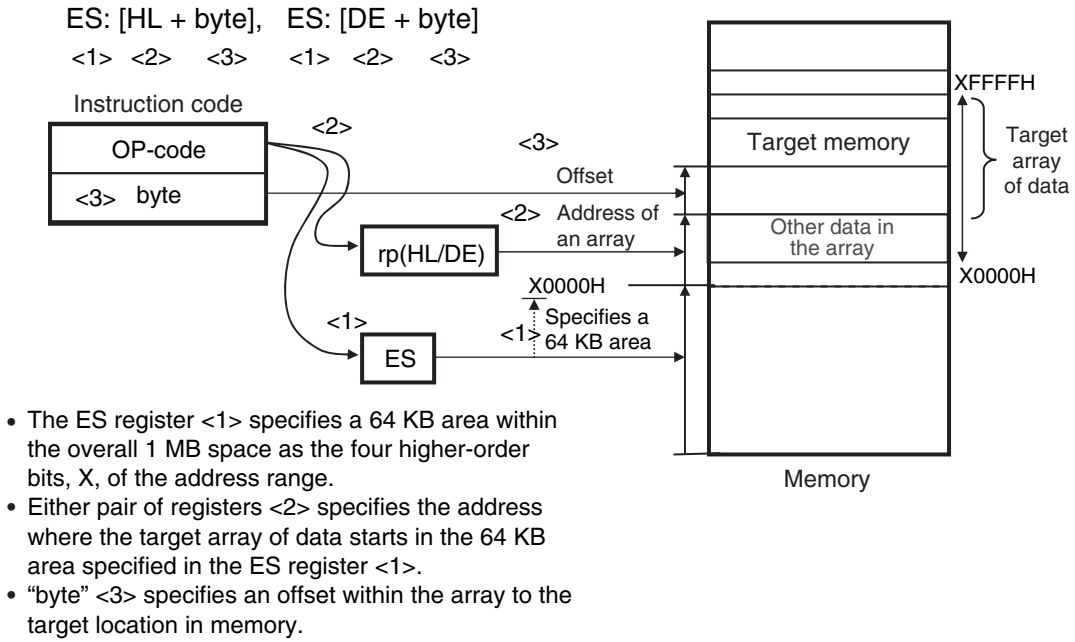


Figure 3-33. Example of ES:word[B], ES:word[C]

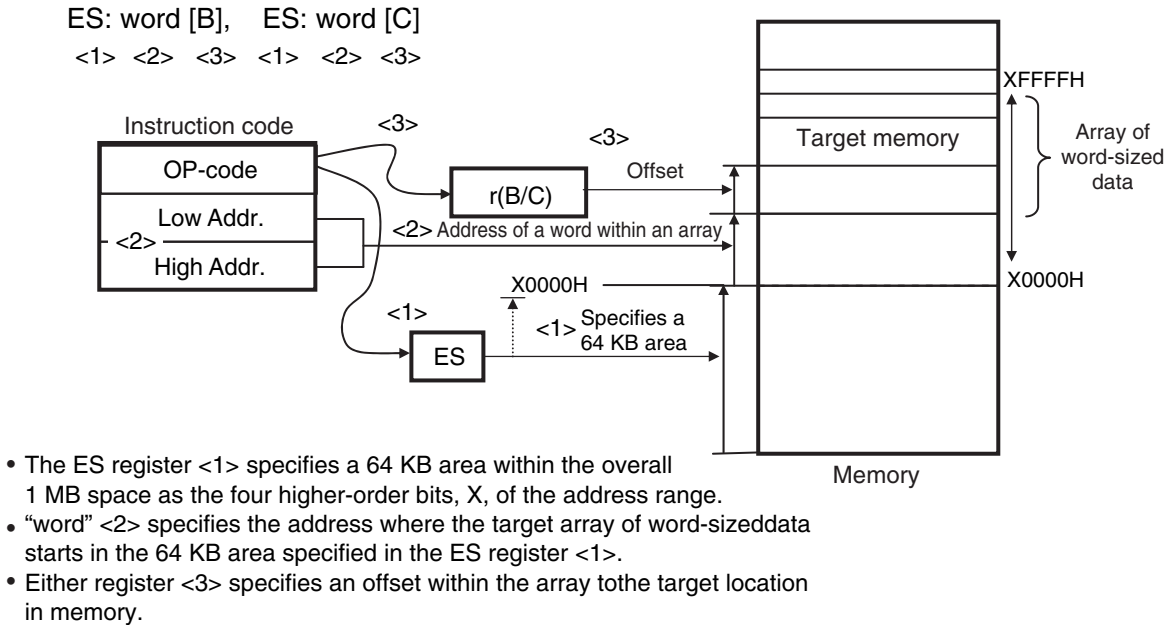
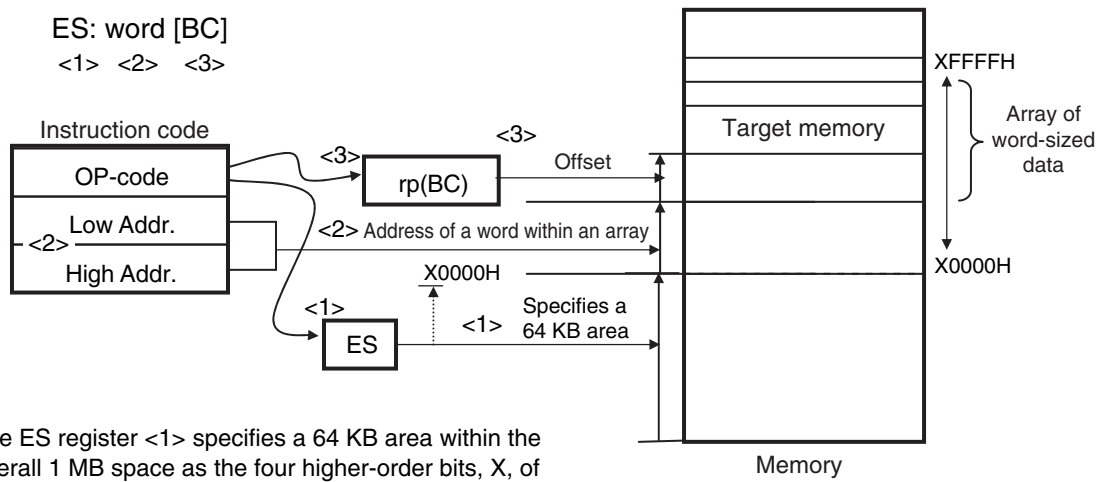


Figure 3-34. Example of ES:word[BC]



- The ES register <1> specifies a 64 KB area within the overall 1 MB 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 KB 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 3-35. Example of [HL+B], [HL+C]

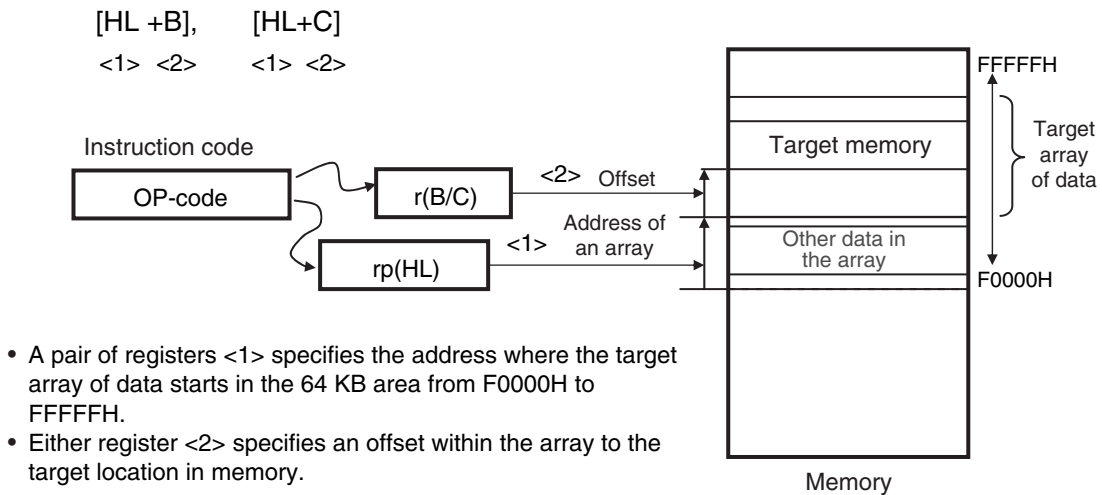
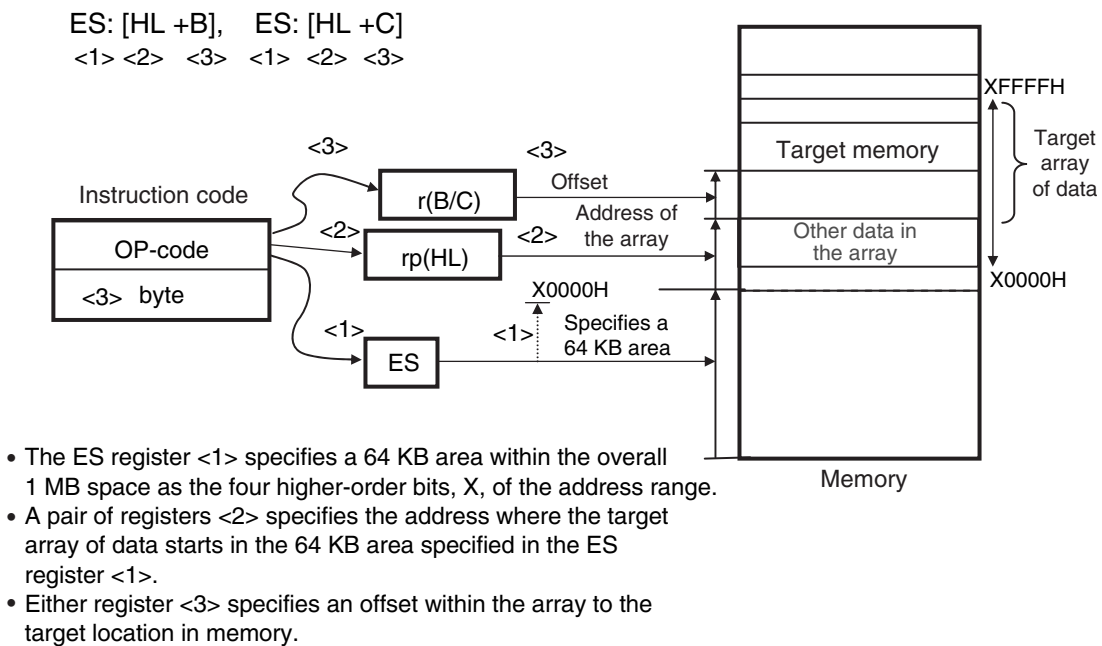


Figure 3-36. Example of ES:[HL+B], ES:[HL+C]



3.4.9 Stack addressing

[Function]

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

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

[Operand format]

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

Each stack operation saves or restores data as shown in Figures 3-37 to 3-42.

Figure 3-37. Example of PUSH rp

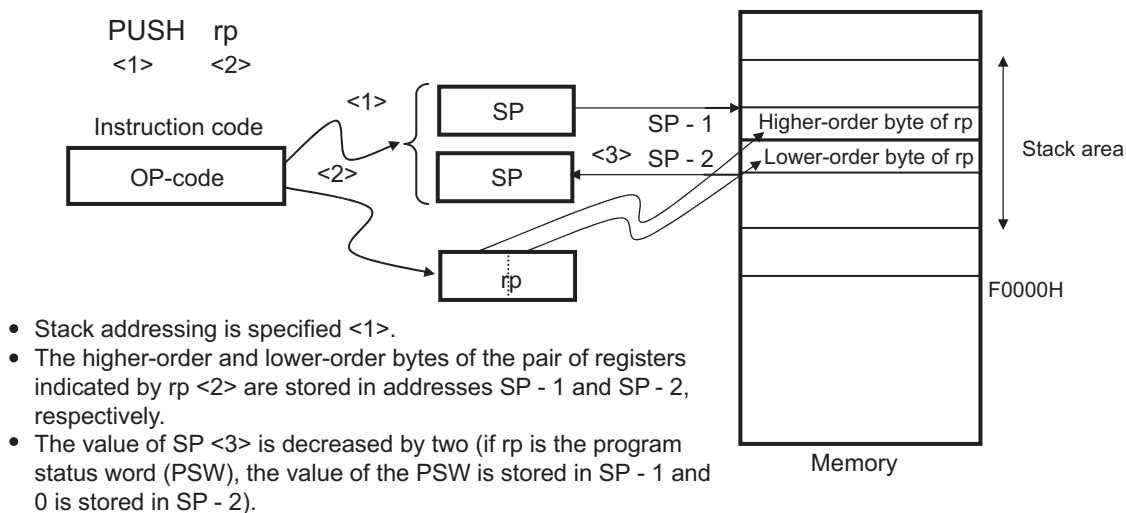
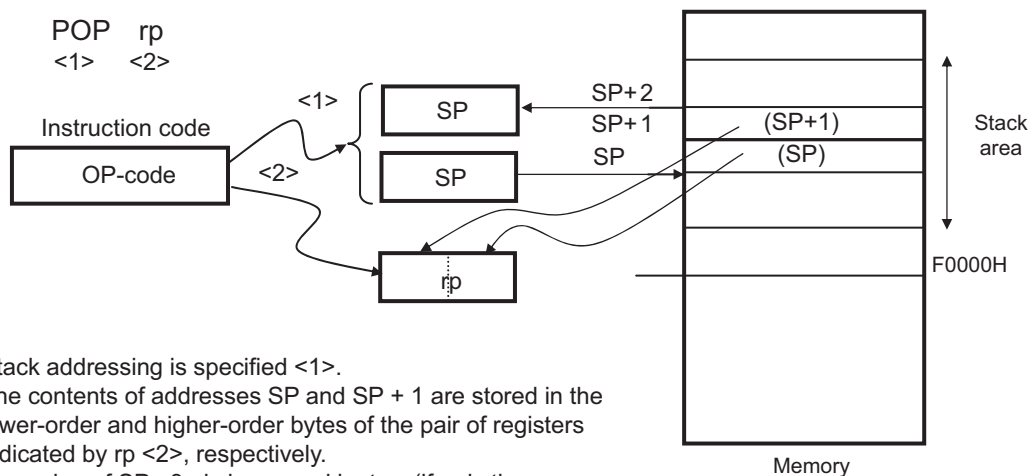


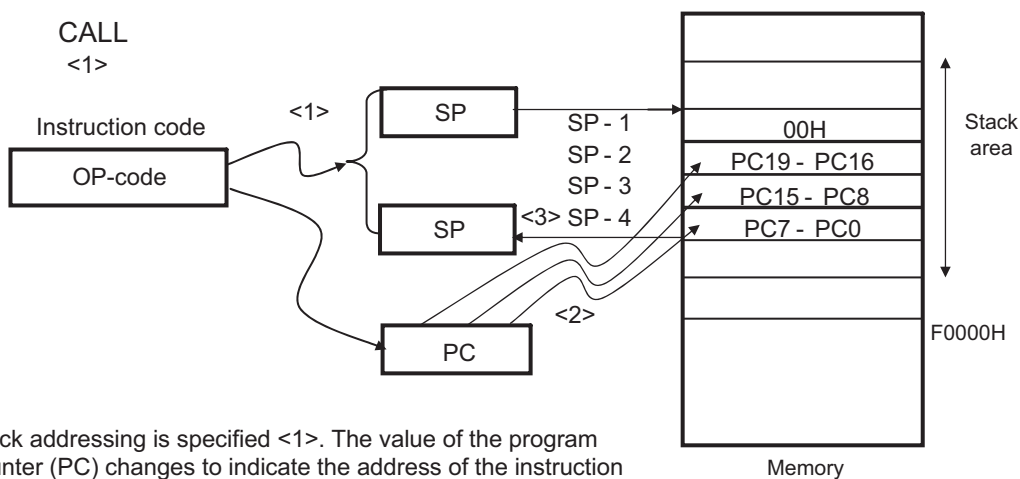


Figure 3-38. Example of POP



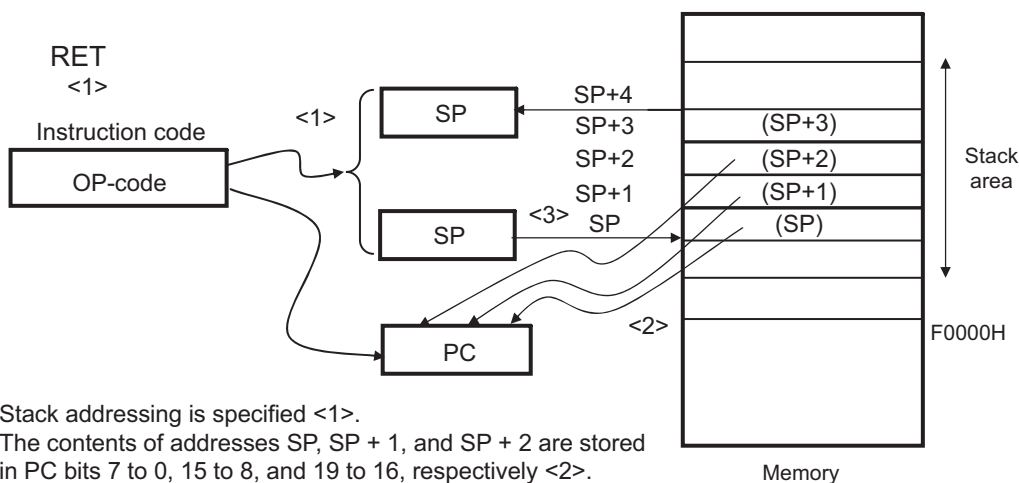
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP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

Figure 3-39. Example of CALL, CALLT



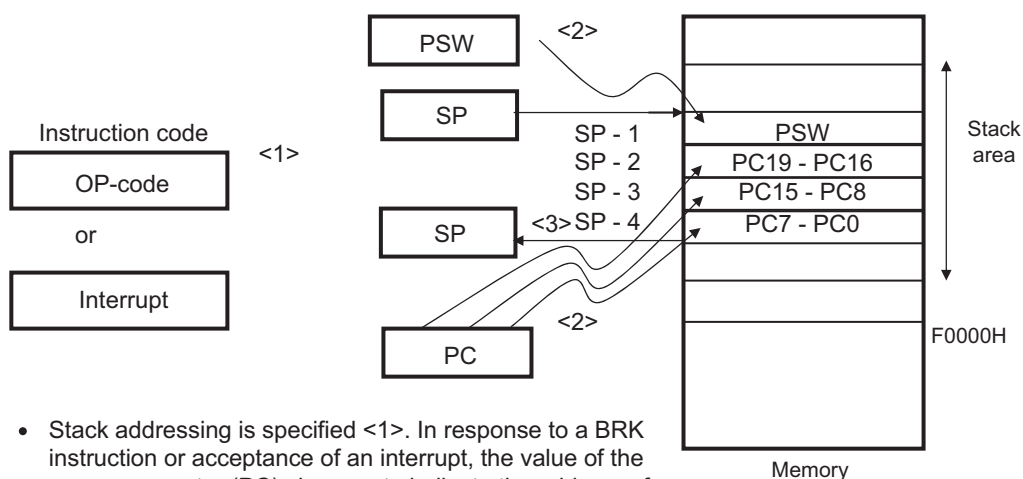
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP - 1, SP - 2, SP - 3, and SP - 4, respectively <2>.
- The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

Figure 3-40. Example of RET



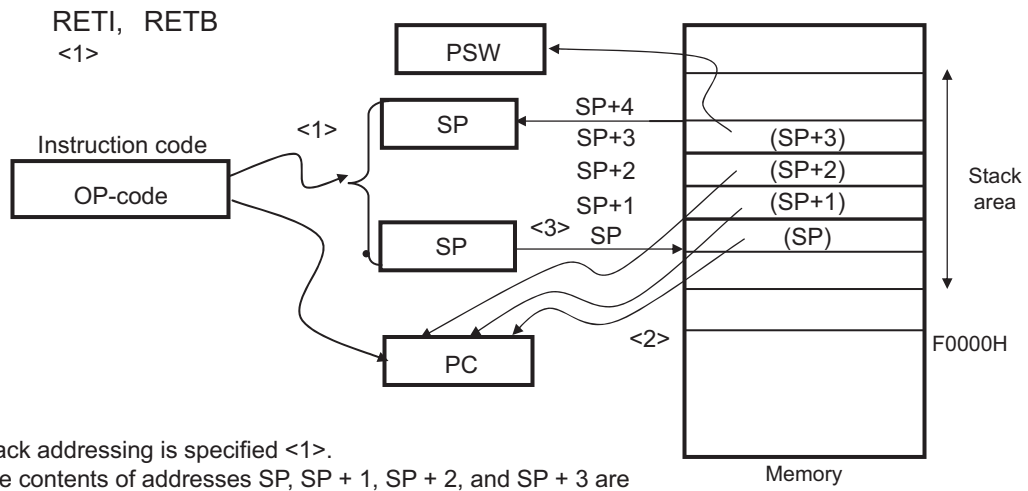
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3-41. 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 3-42. 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/L13 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) Port registers (P0 to P7, P12, P13) Pull-up resistor option registers (PU0 to PU5, PU7, PU12, PU13) Port input mode registers (PIM0, PIM1, PIM3 to PIM5) Port output mode registers (POM0, POM1, POM3 to POM5, POM13) Port mode control registers (PMC1, PMC2, PMC4) A/D port configuration register (ADPC) Peripheral I/O redirection register (PIOR) LCD port function registers (PFSEG0 to PFSEG6) LCD input switch control register (ISCLCD)
Port	<ul style="list-style-type: none"> <li>• 64-pin products                Total: 49 (CMOS I/O: 42 (N-ch open drain I/O [<math>V_{DD}</math> tolerance]: 12), CMOS input: 5, N-ch open drain I/O: 2)</li> <li>• 80-pin products                Total: 65 (CMOS I/O: 58 (N-ch open drain I/O [<math>V_{DD}</math> tolerance]: 18), CMOS input: 5, N-ch open drain I/O: 2)</li> </ul>

#### 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 to P07 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 P03, P05 and P06 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 0 (PIM0).

Output from the P00 and P04 to P07 pins can be specified as N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 0 (POM0).

To use all pins as digital I/O port pins, set them to be used as port (other than segment output) by using LCD port function registers 5, 6 (PFSEG5, PFSEG6) (can be specified in 1-bit units).

This port can also be used for programming UART output, segment output of LCD controller/driver, serial interface data I/O, and clock I/O, timer I/O, clock/buzzer output, external interrupt request input, and comparator output.

Reset signal generation sets port 0 to the digital input invalid mode<sup>Note</sup>.

**Note** “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

#### 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 P10 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).

Input to the P16 and P17 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 1 (PIM1).

Output from the P16 and P17 pins can be specified as N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 1 (POM1).

To use the P10 to P13 pins as digital I/O port pins, set them in the digital I/O mode by using port mode control register 1 (PMC1) and to be used as port (other than segment output) by using LCD port function register 4 (PFSEG4) (can be specified in 1-bit units).

To use the P14 to P17 pins as digital I/O port pins, set them to be used as port (other than segment output) by using LCD port function register 5 (PFSEG5) (can be specified in 1-bit units).

This port can also be used for A/D converter analog input, segment output of LCD controller/driver, serial interface data I/O, clock I/O, timer I/O, and programming UART input.

Reset signal generation sets the P10 to P13 pins to analog input and the P14 to P17 pins to the digital input invalid mode<sup>Note</sup>.

**Note** “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

**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). When the P22 to P27 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 2 (PU2).

This port can also be used for A/D converter analog input, (+side and –side) reference voltage input, and segment output of LCD controller/driver.

To use P20/ANI1 and P21/ANI0 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC). Use these pins starting from the upper bit.

To use P20/ANI1 and P21/ANI0 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/ANI1 and P21/ANI0 Pins**

ADPC Register	PM2 Register	ADS Register	P20/ANI1 and P21/ANI0 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.	

To use the P22 to P27 pins as digital I/O port pins, set them in the digital I/O mode by using port mode control register 2 (PMC2) and to be used as port (other than segment output) by using LCD port function registers 3, 4 (PFSEG3, PFSEG4) (can be specified in 1-bit units).

P20 to P27 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 to P35 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).

Input to the P34 pin can be specified through a normal input buffer or a TTL input buffer using port input mode register 3 (PIM3).

Output from the P35 pins can be specified as N-ch open-drain output (V<sub>DD</sub> tolerance) using port output mode register 3 (POM3).

To use the P30 to P35 pins as digital I/O port pins, set them to be used as port (other than segment output) by using LCD port function registers 2, 3 (PFSEG2, PFSEG3) (can be specified in 1-bit units).

This port can also be used for external interrupt request input, real-time clock 2 correction clock output, timer I/O, remote controller output, segment output of LCD controller/driver, and serial interface data I/O.

Reset signal generation sets port 3 to the digital input invalid mode<sup>Note</sup>.

**Note** “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

#### 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 to P47 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).

Input to the P43, P44, P46, and P47 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 4 (PIM4).

Output from the P42 to P44, P46, and P47 pins can be specified as N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 4 (POM4).

To use the P45 pin as a digital I/O port pin, set it to be used as port (other than segment output) by using bit 2 (PFDEG) of LCD port function register 3 (PFSEG3).

To use the P46 and P47 pins as digital I/O port pins, set them to be used as port (other than segment output) by using LCD port function register 3 (PFSEG3) (can be specified in 1-bit units).

This port can also be used for serial interface data I/O, clock I/O, data I/O for a flash memory programmer/debugger, timer I/O, comparator reference voltage input, and comparator analog voltage input, and segment output of LCD controller/driver.

Reset signal generation sets the P40 to P44 pins to input mode and the P45 to P47 pins to the digital input invalid mode<sup>Note</sup>.

**Note** “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

#### 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 to P57 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 P53 and P55 pins 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 P53 to P56 pin can be specified as N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 5 (POM5).

To use all pins as digital I/O port pins, set them to be used as port (other than segment output) by using LCD port function registers 0, 1 (PFSEG0, PFSEG1) (can be specified in 1-bit units).

This port can also be used for segment output of LCD controller/driver, timer I/O, and external interrupt request input.

Reset signal generation sets port 5 to the digital input invalid mode<sup>Note</sup>.

**Note** “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

#### 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 and P61 pins is N-ch open-drain output (6 V tolerance).

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

To use all pins as digital I/O port pins, set them to be used as port (other than segment output) by using LCD port function registers 1, 2 (PFSEG1, PFSEG2) (can be specified in 1-bit units).

This port can also be used for key interrupt input, timer I/O, segment output of LCD controller/driver, Reset signal generation sets port 7 to the digital input invalid mode<sup>Note</sup>.

**Note** "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

#### 4.2.9 Port 12

P125 to P127 are a 3-bit 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.

To use the P125 to P127 pins as digital I/O pins, set them in the digital input valid mode by using the LCD input switch control register (ISCLCD) (can be specified in 1-bit units).

This port can also be used for connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, external clock input for subsystem clock, remote control output, timer I/O, connecting a capacitor for LCD controller/driver, and power supply voltage pin for driving the LCD.

Reset signal generation sets P121 to P124 to input mode. P125 to P127 are set in the digital invalid mode<sup>Note</sup>.

**Note** "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

#### 4.2.10 Port 13

P130 is an I/O port with an output latch. P130 can be set to the input mode or output mode in 1-bit units using port mode register 13 (PM13). 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 13 (PU13).

P137 is a 1-bit input-only port.

To use the P130 pin as a digital I/O port pin, set it to be used as port (other than segment output) by using LCD port function register 3 (PFSEG3).

This port can also be used for external interrupt request input, serial interface data output, segment output of LCD controller/driver, and serial interface data output,

Reset signal generation sets the P137 pin to input mode and the P130 pin to the digital input invalid mode<sup>Note</sup>.

**Note** "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.



### 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)
- LCD port function registers (PFSEG0 to PFSEG6)
- LCD input switch control register (ISCLCD)

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

Table 4-3. PMxx, Pxx, PUxx, PIMxx, POMxx, and PMCxx Registers and Bits Mounted on Each Product (1/3)

Port		Bit Name						80 Pin	64 Pin
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register	PMCxx Register		
Port 0	0	PM00	P00	PU00	–	POM00	–	√	√
	1	PM01	P01	PU01	–	–	–	√	√
	2	PM02	P02	PU02	–	–	–	√	√
	3	PM03	P03	PU03	PIM03	–	–	√	√
	4	PM04	P04	PU04	–	POM04	–	√	√
	5	PM05	P05	PU05	PIM05	POM05	–	√	√
	6	PM06	P06	PU06	PIM06	POM06	–	√	√
	7	PM07	P07	PU07	–	POM07	–	√	√
Port 1	0	PM10	P10	PU10	–	–	PMC10	√	√
	1	PM11	P11	PU11	–	–	PMC11	√	√
	2	PM12	P12	PU12	–	–	PMC12	√	√
	3	PM13	P13	PU13	–	–	PMC13	√	√
	4	PM14	P14	PU14	–	–	–	√	√
	5	PM15	P15	PU15	–	–	–	√	√
	6	PM16	P16	PU16	PIM16	POM16	–	√	√
	7	PM17	P17	PU17	PIM17	POM17	–	√	√
Port 2	0	PM20	P20	–	–	–	–	√	√
	1	PM21	P21	–	–	–	–	√	√
	2	PM22	P22	PU22	–	–	PMC22	√	√
	3	PM23	P23	PU23	–	–	PMC23	√	–
	4	PM24	P24	PU24	–	–	PMC24	√	–
	5	PM25	P25	PU25	–	–	PMC25	√	–
	6	PM26	P26	PU26	–	–	PMC26	√	√
	7	PM27	P27	PU27	–	–	PMC27	√	√
Port 3	0	PM30	P30	PU30	–	–	–	√	√
	1	PM31	P31	PU31	–	–	–	√	√
	2	PM32	P32	PU32	–	–	–	√	√
	3	PM33	P33	PU33	–	–	–	√	√
	4	PM34	P34	PU34	PIM34	–	–	√	–
	5	PM35	P35	PU35	–	POM35	–	√	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–

Table 4-3. PMxx, Pxx, PUxx, PIMxx, POMxx, and PMCxx Registers and Bits Mounted on Each Product (2/3)

Port		Bit Name						80 Pin	64 Pin
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register	PMCxx Register		
Port 4	0	PM40	P40	PU40	–	–	–	√	√
	1	PM41	P41	PU41	–	–	–	√	–
	2	PM42	P42	PU42	–	POM42	PMC42	√	√
	3	PM43	P43	PU43	PIM43	POM43	PMC43	√	√
	4	PM44	P44	PU44	PIM44	POM44	PMC44	√	√
	5	PM45	P45	PU45	–	–	PMC45	√	√
	6	PM46	P46	PU46	PIM46	POM46	–	√	–
	7	PM47	P47	PU47	PIM47	POM47	–	√	–
Port 5	0	PM50	P50	PU50	–	–	–	√	–
	1	PM51	P51	PU51	–	–	–	√	–
	2	PM52	P52	PU52	–	–	–	√	√
	3	PM53	P53	PU53	PIM53	POM53	–	√	√
	4	PM54	P54	PU54	–	POM54	–	√	√
	5	PM55	P55	PU55	PIM55	POM55	–	√	–
	6	PM56	P56	PU56	–	POM56	–	√	–
	7	PM57	P57	PU57	–	–	–	√	√
Port 6	0	PM60	P60	–	–	–	–	√	√
	1	PM61	P61	–	–	–	–	√	√
	2	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–
	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	–	–	–	√	√
	5	PM75	P75	PU75	–	–	–	√	√
	6	PM76	P76	PU76	–	–	–	√	√
	7	PM77	P77	PU77	–	–	–	√	√

Table 4-3. PMxx, Pxx, PUxx, PIMxx, POMxx, and PMCxx Registers and Bits Mounted on Each Product (3/3)

Port		Bit Name						80 Pin	64 Pin
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register	PMCxx Register		
Port 12	0	–	–	–	–	–	–	–	–
	1	–	P121	–	–	–	–	√	√
	2	–	P122	–	–	–	–	√	√
	3	–	P123	–	–	–	–	√	√
	4	–	P124	–	–	–	–	√	√
	5	PM125	P125	PU125	–	–	–	√	√
	6	PM126	P126	PU126	–	–	–	√	√
	7	PM127	P127	PU127	–	–	–	√	√
Port 13	0	PM130	P130	PU130	–	POM130	–	√	–
	1	–	–	–	–	–	–	–	–
	2	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	P137	–	–	–	–	√	√

**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	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	PM35	PM34	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM12	PM127	PM126	PM125	1	1	1	1	1	FFF2CH	FFH	R/W
PM13	1	1	1	1	1	1	1	PM130	FFF2DH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 7, 12, 13; 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<sup>Note</sup>.

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 P10 to P13, P20 to P27 are set up as analog inputs of the A/D converter, or P42 to P45 are set up as analog inputs of the comparator, 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	P07	P06	P05	P04	P03	P02	P01	P00	FFF00H	00H (output latch)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	0	0	P35	P34	P33	P32	P31	P30	FFF03H	00H (output latch)	R/W
P4	P47	P46	P45	P44	P43	P42	P41	P40	FFF04H	00H (output latch)	R/W
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	0	0	P61	P60	FFF06H	00H (output latch)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W
P12	P127	P126	P125	P124	P123	P122	P121	0	FFF0CH	Undefined	R/W <sup>Note 1</sup>
P13	P137	0	0	0	0	0	0	P130	FFF0DH	<b>Note 2</b>	R/W <sup>Note 1</sup>

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, 13 ; 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 normal output mode (POM<sub>m</sub> = 0) and input mode (PM<sub>m</sub> = 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 and analog setting (PMC = 1, ADPC = 1), 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).

**Caution** When a port with the PIM<sub>n</sub> register is input from different potential device to TTL buffer, pull up to the power supply of the different potential device via an external pull-up resistor by setting PUm<sub>n</sub> = 0.

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	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU2	PU27	PU26	PU25	PU24	PU23	PU22	0	0	F0032H	00H	R/W
PU3	0	0	PU35	PU34	PU33	PU32	PU31	PU30	F0033H	00H	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	F0034H	01H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU12	PU127	PU126	PU125	0	0	0	0	0	F003CH	00H	R/W
PU13	0	0	0	0	0	0	0	PU130	F003DH	00H	R/W

PU <sub>m</sub> <sub>n</sub>	P <sub>m</sub> n pin on-chip pull-up resistor selection (m = 0 to 5, 7, 12, 13 ; 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 (PIMxx)

These registers set the input buffer in 1-bit units.

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

Port input mode 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	PIM06	PIM05	0	PIM03	0	0	0	F0040H	00H	R/W
PIM1	PIM17	PIM16	0	0	0	0	0	0	F0041H	00H	R/W
PIM3	0	0	0	PIM34	0	0	0	0	F0043H	00H	R/W
PIM4	PIM47	PIM46	0	PIM44	PIM43	0	0	0	F0044H	00H	R/W
PIM5	0	0	PIM55	0	PIM53	0	0	0	F0045H	00H	R/W

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

These registers set the output mode 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 SDA10 pins during simplified I<sup>2</sup>C communication with an external device of the same potential.

In addition, POMxx register is set with PUxx register, whether or not to use the on-chip pull-up resistor.

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 (POMmn = 1) 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	POM07	POM06	POM05	POM04	0	0	0	POM00	F0050H	00H	R/W
POM1	POM17	POM16	0	0	0	0	0	0	F0051H	00H	R/W
POM3	0	0	POM35	0	0	0	0	0	F0053H	00H	R/W
POM4	POM47	POM46	0	POM44	POM43	POM42	0	0	F0054H	00H	R/W
POM5	0	POM56	POM55	POM54	POM53	0	0	0	F0055H	00H	R/W
POM13	0	0	0	0	0	0	0	POM130	F005DH	00H	R/W
POMmn	Pmn pin output mode selection (m = 0, 1, 3 to 5, 13 ; n = 0, 2 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 (PMCxx)**

These registers set the 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 (only PMC4 is set to 00H).

**Figure 4-6. Format of Port Mode Control Register**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC1	1	1	1	1	PMC13	PMC12	PMC11	PMC10	F0061H	FFH	R/W
PMC2	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	1	1	F0062H	FFH	R/W
PMC4	0	0	PMC45	PMC44	PMC43	PMC42	0	0	F0064H	00H	R/W

PMCmn	Pmn pin digital I/O/analog input selection (m = 1, 2, 4; n = 0 to 7)	
0	Digital I/O (alternate function other than analog input)	
1	Analog input	

- Cautions 1. Select input mode by using port mode registers 1, 2, and 4 (PM1, PM2, and PM4) for the ports which are set by the PMCxx register as analog input.**
- 2. Do not set the pin set by the PMC 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 ANI0/P21, ANI1/P20 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	0	0	ADPC1	ADPC0

ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching	
		ANI1/P20	ANI0/P21
0	0	A	A
0	1	D	D
1	0	D	A
1	1	Setting prohibited	

- Cautions**
1. Set the port to analog input by ADPC register to the input mode by using port mode register 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  $AV_{REFM}$ , set ANI0 and ANI1 to analog input and set the port mode register to the input mode.

### 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, can be changed the settings for redirection until its function enable operation.

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

Reset signal generation sets this register to 00H.

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

Address: F0077H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIOR	0	0	0	PIOR4	PIOR3	PIOR2	PIOR1	PIOR0

Bit	Function	80-pin		64-pin	
		Setting value		Setting value	
		0	1	0	1
PIOR4	INTP7	P02	P43	P02	P43
	INTP5	P55	P01	P01	–
PIOR3	PCLBUZ0	P02	P07	P02	P07
PIOR2	TxD1	P07	P42	P07	P42
	RxD1	P06	P43	P06	P43
	SCL10	P05	P44	P05	P44
	SDA10	P06	P43	P06	P43
	SI10	P06	P43	P06	P43
	SO10	P07	P42	P07	P42
	SCK10	P05	P44	P05	P44
PIOR1	TxD0	P00	P130	This area cannot be used. Be set to 0 (default value).	
	RxD0	P17	P47		
	SCL00	P16	P46		
	SDA00	P17	P47		
	SI00	P17	P47		
	SO00	P00	P130		
	SCK00	P16	P46		
PIOR0	TI00/TO00	P52	P40	P52	P40
	TI01/TO01	P32	P60	P32	P60
	TI02/TO02	P54	P61	P54	P61
	TI03/TO03/ REMOOUT	P30	P127	P30	P127
	TI04/TO04	P14	P126	P14	P126
	TI05/TO05	P42	P01	P42	P01
	TI06/TO06	P56	P125	P125	–
	TI07/TO07	P15	P41	P15	–

**Remark** –: These functions are not available for use.

#### 4.3.9 LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)

These registers set whether to use pins P00 to P07, P10 to P17, P22 to P27, P30 to P35, P46, P47, P50 to P57, P70 to P77, and P130 as port pins (other than segment output pins) or segment output, and whether to use P45 as a port pin or an analog input pin.

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

Reset signal generation sets these registers to FFH (PFSEG0 is set to F0H, and PFSEG6 is set to 0FH).

**Remark** The correspondence between the segment output pins (SEGxx) and the PFSEG register (PFSEGxx bits) and the existence of SEGxx pins in each product are shown in **Table 4-4 Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits)**.

**Figure 4-9. Format of LCD Port Function Registers 0 to 6 (PFSEG0 to PFSEG6)**

Address: F0300H After reset: F0H R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG0	PFSEG07	PFSEG06	PFSEG05	PFSEG04	0	0	0	0

Address: F0301H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG1	PFSEG15	PFSEG14	PFSEG13	PFSEG12	PFSEG11	PFSEG10	PFSEG09	PFSEG08

Address: F0302H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG2	PFSEG23	PFSEG22	PFSEG21	PFSEG20	PFSEG19	PFSEG18	PFSEG17	PFSEG16

Address: F0303H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG3	PFSEG30	PFSEG29	PFSEG28	PFSEG27	PFSEG26	PFDEG	PFSEG25	PFSEG24

Address: F0304H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG4	PFSEG38	PFSEG37	PFSEG36	PFSEG35	PFSEG34	PFSEG33	PFSEG32	PFSEG31

Address: F0305H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG5	PFSEG46	PFSEG45	PFSEG44	PFSEG43	PFSEG42	PFSEG41	PFSEG40	PFSEG39

Address: F0306H After reset: 0FH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG6	0	0	0	0	PFSEG50	PFSEG49	PFSEG48	PFSEG47

PFSEGxx (xx = 04 to 50)	Port (other than segment output)/segment outputs specification of Pmn pins (mn = 00 to 07, 10 to 17, 22 to 27, 30 to 35, 46, 47, 50 to 57, 70 to 77, 130)
0	Used the Pmn pin as port (other than segment output)
1	Used the Pmn pin as segment output

PFDEG	Specification of port/analog input (IVREF0) for P45 pin
0	Used the Pmn pin as port
1	Used the Pmn pin as analog input (IVREF0)

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

**Remark** To use the Pmn pins as segment output pins (PFSEGxx = 1), be sure to set the PUm<sub>n</sub> bit of the PUm register, POM<sub>n</sub> bit of the POM<sub>n</sub> register, and PIM<sub>n</sub> bit of the PIM<sub>n</sub> register to "0".

**Table 4-4. Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits)**

Bit Name of PFSEG Register	Corresponding SEGxx Pins	Alternate Port	80-pin	64-pin
PFSEG04	SEG4	P50	√	–
PFSEG05	SEG5	P51	√	–
PFSEG06	SEG6	P52	√	√
PFSEG07	SEG7	P53	√	√
PFSEG08	SEG8	P54	√	√
PFSEG09	SEG9	P55	√	–
PFSEG10	SEG10	P56	√	–
PFSEG11	SEG11	P57	√	√
PFSEG12	SEG12	P70	√	√
PFSEG13	SEG13	P71	√	–
PFSEG14	SEG14	P72	√	–
PFSEG15	SEG15	P73	√	–
PFSEG16	SEG16	P74	√	√
PFSEG17	SEG17	P75	√	√
PFSEG18	SEG18	P76	√	√
PFSEG19	SEG19	P77	√	√
PFSEG20	SEG20	P30	√	√
PFSEG21	SEG21	P31	√	√
PFSEG22	SEG22	P32	√	√
PFSEG23	SEG23	P33	√	√
PFSEG24	SEG24	P34	√	–
PFSEG25	SEG25	P35	√	–
PFSEG26	SEG26	P46	√	–
PFSEG27	SEG27	P47	√	–
PFSEG28	SEG28	P130	√	–
PFSEG29	SEG29	P22	√	√
PFSEG30	SEG30	P23	√	–
PFSEG31	SEG31	P24	√	–
PFSEG32	SEG32	P25	√	–
PFSEG33	SEG33	P26	√	√
PFSEG34	SEG34	P27	√	√
PFSEG35	SEG35	P10	√	√
PFSEG36	SEG36	P11	√	√
PFSEG37	SEG37	P12	√	√
PFSEG38	SEG38	P13	√	√
PFSEG39	SEG39	P14	√	√
PFSEG40	SEG40	P15	√	√
PFSEG41	SEG41	P16	√	√
PFSEG42	SEG42	P17	√	√
PFSEG43	SEG43	P00	√	√
PFSEG44	SEG44	P01	√	√
PFSEG45	SEG45	P02	√	√
PFSEG46	SEG46	P03	√	√
PFSEG47	SEG47	P04	√	√
PFSEG48	SEG48	P05	√	√
PFSEG49	SEG49	P06	√	√
PFSEG50	SEG50	P07	√	√

#### 4.3.10 LCD input switch control register (ISCLCD)

This register sets whether to use pins P125 to P127 as port pins (other than LCD function pins) or LCD function pins ( $V_{L3}$ , CAPL, CAPH).

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

Reset signal generation sets this register to 00H.

**Figure 4-10. Format of LCD Input Switch Control Register (ISCLCD)**

Address: F0308H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISCLCD	0	0	0	0	0	0	ISCVL3	ISCCAP

ISCVL3	Control of Schmitt trigger buffer of $V_{L3}$ /P125 pin
0	Makes digital input invalid (used as LCD function pin ( $V_{L3}$ ))
1	Makes digital input valid

ISCCAP	Control of Schmitt trigger buffer of CAPL/P126 and CAPH/P127 pins
0	Makes digital input invalid (used as LCD function pins (CAPL, CAPH))
1	Makes digital input valid

**Caution** If ISCVL3 bit = 0 and ISCCAP bit = 0, set the corresponding port control registers as follows:

**PU127 bit of PU12 register = 0, P127 bit of P12 register = 0**

**PU126 bit of PU12 register = 0, P126 bit of P12 register = 0**

**PU125 bit of PU12 register = 0, P125 bit of P12 register = 0**



## 4.4 Port Function Operations

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

### 4.4.1 Writing to I/O port

#### (1) Output mode

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

#### (2) Input mode

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

### 4.4.2 Reading from I/O port

#### (1) Output mode

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

#### (2) Input mode

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

### 4.4.3 Operations on I/O port

#### (1) Output mode

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

#### (2) Input mode

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

#### 4.4.4 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 port input mode registers 0, 1, 3, 4, and 5 (PIM0, PIM1, PIM3, PIM4, 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 port output mode registers 0, 1, 3, 4, 5, and 13 (POM0, POM1, POM3, POM4, POM5, and POM13) on a bit-by-bit basis to enable normal output (CMOS)/N-ch open drain ( $V_{DD}$  tolerance) switching.

Following, describes the connection of a serial interface.

##### (1) Setting procedure when using input pins of UART0 to UART3, CSI00, and CSI10 functions for the TTL input buffer

In case of UART0:	P17 (P47)
In case of UART1:	P06 (P43)
In case of UART2:	P03
In case of UART3:	P34
In case of CSI00:	P17, P16 (P47, P46)
In case of CSI10:	P06, P05 (P44, P43)

**Remark** Functions in parentheses can be assigned by setting 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, PIM1, PIM3, and PIM4 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/CSI mode.

##### (2) Setting procedure when using output pins of UART0 to UART3, CSI00, and CSI10 functions in N-ch open-drain output mode

In case of UART0:	P00 (P130)
In case of UART1:	P07 (P42)
In case of UART2:	P04
In case of UART3:	P35
In case of CSI00:	P00, P16 (P130, P46)
In case of CSI10:	P07, P05 (P42, P44)

**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 changes to 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, POM3, POM4, and POM13 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/CSI mode.
- <6> Set the output mode by manipulating the PM0, PM1, PM3, PM4, and PM13 registers.  
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 pins of IIC00 and IIC10 functions with a different potential (1.8 V, 2.5 V, 3 V)**

In case of IIC00: P16, P17 (P46, P47)

In case of IIC10: P05, P06 (P44, P43)

**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, and POM4 registers to 1 to set the N-ch open drain output ( $V_{DD}$  tolerance) mode.
- <5> Set the corresponding bit of the PIM0, PIM1, and PIM4 registers to 1 to switch 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<sup>2</sup>C mode.
- <7> Set the corresponding bit of the PM0, PM1, and PM4 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

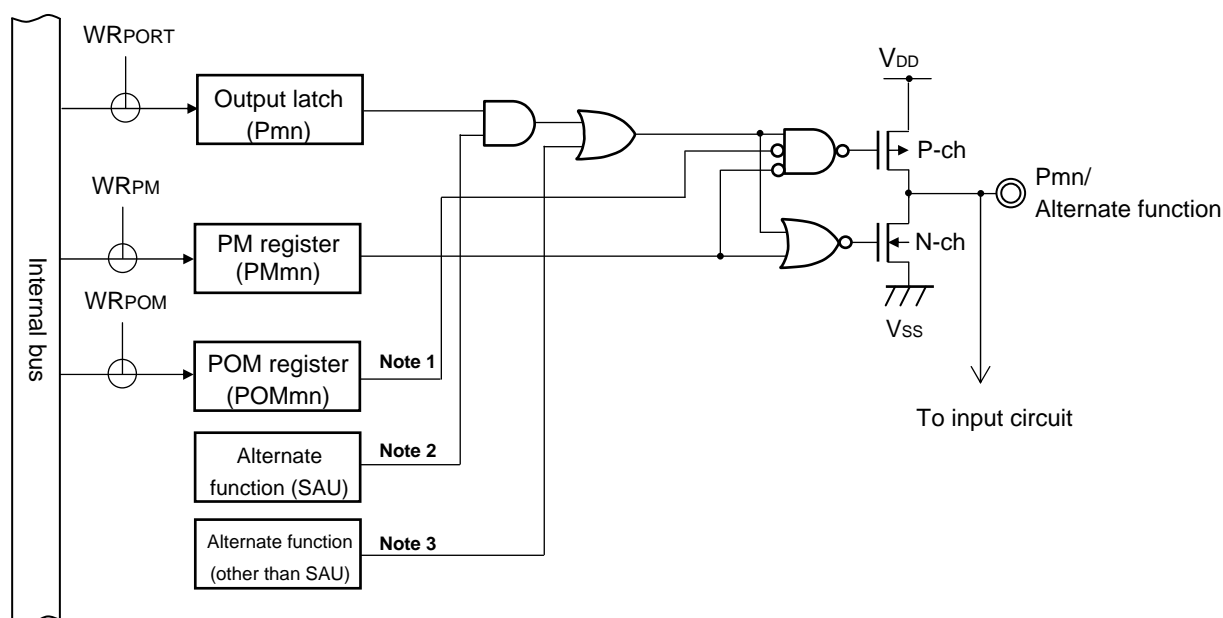
#### 4.5.1 Basic concept when using alternate function

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

Also, for a pin also assigned to be used for analog input and segment output, use the LCD port function register (PFSEGx) and port mode control register (PMCxx) specify whether to use the pin for analog input, segment output, or digital input/output.

Figure 4-11 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, RTC2, 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 4-5.

Figure 4-11. 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, 13); n: Bit number (n = 0 to 7)

Table 4-5. Concept of Basic Settings

Output Function of Used Pin	Output Settings of Unused Alternate Function		
	Port Function	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) <sup>Note</sup>

**Note** 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 (SO<sub>p</sub>/Tx<sub>Dq</sub>) of SAU is not used)

When the serial output (SO<sub>p</sub>/Tx<sub>Dq</sub>) 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* (SO<sub>Em</sub>) which corresponds to the unused output to 0 (output disabled) and set the SO<sub>mn</sub> bit in serial output register *m* (SO<sub>m</sub>) 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* (SE<sub>mn</sub>) in serial channel enable status register *m* (SE<sub>m</sub>) to 0 (operation stopped state), set the bit in serial output enable register *m* (SO<sub>Em</sub>) which corresponds to the unused output to 0 (output disabled), and set the SO<sub>mn</sub> and CKO<sub>mn</sub> bits in serial output register *m* (SO<sub>m</sub>) 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 TO<sub>mn</sub> 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 *n*0 (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.

- (6)  $REMOOUT = 0$  (setting when remote control output is not used)

When the REMOOUT output of TAU is not used, set the bit in timer output enable register 0 (TOE0) of channels 2, 3, 4, and 5 to 0 (output disabled), and set the bit in timer output register 0 (TO0) to 0 (Low). This is the same setting as the initial state.

- (7) VCOUTn = 0 (setting when VCOUTn is not used)

When VCOUTn of comparator is not used, set the bits 5 and 1 in the comparator output control register (COMPOCR) to 0 (VCOUTn pin of comparator n output disabled). This is the same setting as the initial state.

- <R> (8) TKBO00 = 0/TKBO01-0, 1, 2 = 0 (setting when 16-bit timer KB20 is not used)

When 16-bit timer KB2 is not used, set the bit 7 in 16-bit timer KB20 operation control register 01 (TKBCTL01) to 0 (timer operation stopped), set the bits 1 and 0 in 16-bit timer KB2 output control register 01 (TKBIOC01) to 0 (timer output disabled), set the bits 1 and 0 in 16-bit timer KB2 output control register 00 (TKBIOC00) to 0 (default level is low level), and set the bits 0 and 1 in forced output stop function control register 2 (TKBPACTL02) to 0 (forced output stop function operation 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 4-6. The registers used to control the port functions should be set as shown in Table 4-6. See the following remark for legends used in Table 4-6.

<b>Remark</b>	–:	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
	PFSEGxx, PFDEG:	LCD port function register

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

**Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (1/15)**

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note 1</sup>	Alternate Function Output		80-pin	64-pin	
	Function Name	I/O							SAU Output Function	Other than SAU			
P00	P00	Input	–	×	–	1	×	0	–	–	√	√	
		Output		0		0	0/1		SO00/TxD0 = 1				
		N-ch open drain output	×	1									
	SO00	Output	PIOR1 = 0 <sup>Note 2</sup>	0/1		0	1						–
	TxD0	Output	PIOR1 = 0 <sup>Note 2</sup>	0/1		0	1						–
	SEG43	Output	×	0		0	0						–
P01	P01	Input	–	–	–	1	×	0		–	–	√	√
		Output	×			0	0/1		TO05 = 0 PCLBUZ1 = 0				
	INTP5 <sup>Note 3</sup>	Input	PIOR4 = 0			1	×		–		×	√	
	(INTP5) <sup>Note 2</sup>	Input	PIOR4 = 1			1	×		–		√	×	
	(TI05)	Input	PIOR0 = 1			1	×		–		√	√	
	(TO05)	Output				0	0		PCLBUZ1 = 0		√	√	
	PCLBUZ1	Output	×			0	0		TO05 = 0		√	√	
	SEG44	Output	×			0	0		–		√	√	
P02	P02	Input	–	–	–	1	×	0	–	–	√	√	
		Output	×			0	0/1			PCLBUZ0 = 0			
	INTP7	Input	PIOR4 = 0			1	×			–			–
	PCLBUZ0	Output	PIOR3 = 0			0	0			–			–
	SEG45	Output	×			0	0			–			–
P03	P03	Input	–	–	–	1	×	0	–	–	√	√	
		Output				0	0/1			VCOU0 = 0			
	RxD2	Input				1	×			–			–
	VCOU0	Output				0	0			–			–
	SEG46	Output				0	0			–			–

- Notes**
1. PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.
  2. 80-pin products only
  3. 64-pin products only



Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (2/15)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note</sup>	Alternate Function Output		80-pin	64-pin	
	Function Name	I/O							SAU Output Function	Other than SAU			
P04	P04	Input	-	×	-	1	×	0	-	-	√	√	
		Output		0		0			0/1	TxD2 = 1			VCOU1 = 0
		N-ch open drain output		1		0			1	-			VCOU1 = 0
	TxD2	Output		0/1		0	1		-	VCOU1 = 0			
	VCOU1	Output		0		0	0		TxD2 = 1	-			
	SEG47	Output		0		0	0		-	-			
P05	P05	Input	-	×	-	1	×	0	-	-	√	√	
		Output	×	0		0			0/1				SCK10/SCL10 = 1
		N-ch open drain output		1		1			×				-
	SCK10	Input	PIOR2 = 0	×		1	×		-				-
		Output		0/1		0	1		-				-
	SCL10	Output	PIOR2 = 0	0/1		0	1		-				-
SEG48	Output	×	0	0	0	1	-						
P06	P06	Input	-	×	-	1	×	0	-	-	√	√	
		Output	×	0		0			0/1				SDA10 = 1
		N-ch open drain output		1		1			×				-
	SI10	Input	PIOR2 = 0	×		1	×		-				-
	RxD1	Input	PIOR2 = 0	×		1	×		-				-
	SDA10	I/O	PIOR2 = 0	1		0	1		-				-
SEG49	Output	×	0	0	0	1	-						
P07	P07	Input	-	×	-	1	×	0	-	-	√	√	
		Output	×	0		0			0/1	SO10/TxD1 = 1			(PCLBUZ0 = 0)
		N-ch open drain output		1		0			1	-			(PCLBUZ0 = 0)
	SO10	Output	PIOR2 = 0	0/1		0	1		-	(PCLBUZ0 = 0)			
	TxD1	Output	PIOR2 = 0	0/1		0	1		-	(PCLBUZ0 = 0)			
	(PCLBUZ0)	Output	PIOR3 = 1	0		0	0		SO10/TxD1 = 1	-			
SEG50	Output	×	0	0	0	1	-						

**Note** PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.

Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (3/15)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note</sup>	Alternate Function Output		80-pin	64-pin
	Function Name	I/O							SAU Output Function	Other than SAU		
P10	P10	Input	-	-	0	1	×	0	-	-	√	√
		Output			0	0	0/1					
	ANI22	Analog input			1	1	×	1				
	SEG35	Output			0	0	0					
P11	P11	Input	-	-	0	1	×	0	-	-	√	√
		Output			0	0	0/1					
	ANI23	Analog input			1	1	×	1				
	SEG36	Output			0	0	0					
P12	P12	Input	-	-	0	1	×	0	-	-	√	√
		Output			0	0	0/1					
	ANI24	Analog input			1	1	×	1				
	SEG37	Output			0	0	0					
P13	P13	Input	-	-	0	1	×	0	-	-	√	√
		Output			0	0	0/1					
	ANI25	Analog input			1	1	×	1				
	SEG38	Output			0	0	0					
P14	P14	Input	-	-	-	1	×	0	-	-	√	√
		Output	×			0	0/1			TO04 = 0		
	TI04	Input	PIOR0 = 0			1	×			-		
	TO04	Output	PIOR0 = 0			0	0			-		
	SEG39	Output	×			0	0			1		
P15	P15	Input	-	-	-	1	×	0	-	-	√	√
		Output	×			0	0/1			TO07 = 0		
	TI07	Input	PIOR0 = 0			1	×			-		
	TO07	Output	PIOR0 = 0			0	0			-		
	SEG40	Output	×			0	0			1		

**Note** PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.

**Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (4/15)**

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note 1</sup>	Alternate Function Output		80-pin	64-pin
	Function Name	I/O							SAU Output Function	Other than SAU		
P16	P16	Input	–	×	–	1	×	0	–	–	√	√
		Output	×	0		0	0/1		SCK00/SCL00 = 1			
		N-ch open drain output		1								
	SCK00	Input	PIOR1 = 0 <sup>Note 2</sup>	×		1	×	–	–			
		Output		0/1		0	1	–	–			
	SCL00	Output	PIOR1 = 0 <sup>Note 2</sup>	0/1		0	1	–	–			
	SEG41	Output	×	0		0	0	1	–			
P17	P16	Input	–	×	–	1	×	0	–	–	√	√
		Output	×	0		0	0/1		SDA00 = 1			
		N-ch open drain output		1								
	SI00	Input	PIOR1 = 0 <sup>Note 2</sup>	×		1	×	–	–			
	RxD0	Input	PIOR1 = 0 <sup>Note 2</sup>	×		1	×	–	–			
	SDA00	I/O	PIOR1 = 0 <sup>Note 2</sup>	1		0	1	–	–			
	SEG42	Output	×	0		0	0	1	–			

- Notes**
1. PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.
  2. 80-pin products only

**Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (5/15)**

Pin Name	Used Function		ADPC	ADM2	PMxx	Pxx	80-pin	64-pin
	Function Name	I/O						
P20	P20	Input	01H/02H	×	1	×	√	√
		Output		×	0	0/1		
	ANI1	Analog input	00H	xx00xx0xB	1	×		
	AV <sub>REFM</sub>	Reference voltage input	00H	xx10xx0xB	1	×		
P21	P21	Input	01H	×	1	×	√	√
		Output		×	0	0/1		
	ANI0	Analog input	00H/02H	00x0xx0xB 10x0xx0xB	1	×		
	AV <sub>REFP</sub>	Reference voltage input	00H/02H	01x0xx0xB	1	×		

Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (6/15)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note</sup>	Alternate Function Output		80-pin	64-pin
	Function Name	I/O							SAU Output Function	Other than SAU		
P22	P22	Input	-	-	0	1	×	0	-	-	√	√
		Output			0	0	0/1					
	ANI16	Analog input			1	1	×	1				
	SEG29	Output			0	0	0					
P23	P23	Input	-	-	0	1	×	0	-	-	√	×
		Output			0	0	0/1					
	ANI17	Analog input			1	1	×	1				
	SEG30	Output			0	0	0					
P24	P24	Input	-	-	0	1	×	0	-	-	√	×
		Output			0	0	0/1					
	ANI18	Analog input			1	1	×	1				
	SEG31	Output			0	0	0					
P25	P25	Input	-	-	0	1	×	0	-	-	√	×
		Output			0	0	0/1					
	ANI19	Analog input			1	1	×	1				
	SEG32	Output			0	0	0					
P26	P26	Input	-	-	0	1	×	0	-	-	√	√
		Output			0	0	0/1					
	ANI20	Analog input			1	1	×	1				
	SEG33	Output			0	0	0					
P27	P27	Input	-	-	0	1	×	0	-	-	√	√
		Output			0	0	0/1					
	ANI21	Analog input			1	1	×	1				
	SEG34	Output			0	0	0					

**Note** PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.

**Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (7/15)**

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note</sup>	Alternate Function Output		80-pin	64-pin	
	Function Name	I/O							SAU Output Function	Other than SAU			
P30	P30	Input	-	-	-	1	x	0	-	-	√	√	
		Output	x			0	0/1			TO03/REMOOUT = 0			
	TI03	Input	PIOR0 = 0			1	x			-			
	TO03	Output	PIOR0 = 0			0	0			-			
	REMOOUT	Output	PIOR0 = 0			0	0			-			
	SEG20	Output	x			0	0			1			-
P31	P31	Input	-	-	-	1	x	0	-	-	√	√	
		Output	-			0	0/1			RTC1HZ = 0			
	INTP3	Input	-			1	x			-			
	RTC1HZ	Output	-			0	0			-			
	SEG21	Output	-			0	0			1			-
P32	P32	Input	-	-	-	1	x	0	-	-	√	√	
		Output	x			0	0/1			TO01 = 0			
	TI01	Input	PIOR0 = 0			1	x			-			
	TO01	Output	PIOR0 = 0			0	0			-			
	SEG22	Output	x			0	0			1			-
P33	P33	Input	-	-	-	1	x	0	-	-	√	√	
		Output	-			0	0/1			-			
	INTP4	Input	-			1	x			-			
	SEG23	Output	-			0	0			1			-
P34	P34	Input	-	-	-	1	x	0	-	-	√	x	
		Output	-			0	0/1			-			
	RxD3	Input	-			1	x			-			
	SEG24	Output	-			0	0			1			-
P35	P35	Input	-	-	-	1	x	0	-	-	√	x	
		Output	-			0	-			TxD3 = 1			
		N-ch open drain output	-			1	-						
	TxD3	Output	-			0/1	0			0/1			-
	SEG25	Output	-			0	0			x			1
P40	P40	Input	-	-	-	1	x	-	-	-	√	√	
		Output	x			0	0/1			(TO00) = 0			
	(TI00)	Input	PIOR0 = 1			1	x			-			
	(TO00)	Output	PIOR0 = 1			0	0			-			

**Note** PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.

Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (8/15)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note</sup>	Alternate Function Output		80-pin	64-pin		
	Function Name	I/O							SAU Output Function	Other than SAU				
P41	P41	Input	-	-	-	1	×	-	-	-	√	×		
		Output	×			0	0/1			(TO07) = 0				
	(TI07)	Input	PIOR0 = 1			1	×			-				
	(TO07)	Output	PIOR0 = 1			0	0			-				
P42	P42	Input	-	×	0	1	×	-	-	-	√	√		
		Output	×	0		0	0			0/1			(SO10/TxD1) = 1	TO05 = 0
		N-ch open drain output		1			-			-				
	IVREF1	Analog input	-	×	1	1	×		-	-				
	(SO10)	Output	PIOR2 = 1	0/1	0	0	1		-	TO05 = 0				
	(TxD1)	Output	PIOR2 = 1	0/1	0	0	1		-	TO05 = 0				
	TI05	Input	PIOR0 = 0	×	0	1	×		-	-				
TO05	Output	PIOR0 = 0	0	0	0	0	(SO10/TxD1) = 1	-						
P43	P43	Input	-	×	0	1	×	-	-	-	√	√		
		Output	×	0		0	0			0/1			(SDA10) = 1	
		N-ch open drain output		1			-			-				
	IVCMP1	Analog input	-	×	1	1	×		-	-				
	(INTP7)	Input	PIOR4 = 1	×	0	1	×		-	-				
	(SI10)	Input	PIOR2 = 1	×	0	1	×		-	-				
	(RxD1)	Input	PIOR2 = 1	×	0	1	×		-	-				
(SDA10)	I/O	PIOR2 = 1	1	0	0	1	-	-						
P44	P44	Input	-	×	0	1	×	-	-	-	√	√		
		Output	×	0		0	0			0/1			(SCK10/SCL10) = 1	
		N-ch open drain output		1			-			-				
	IVCMP0	Analog input	-	×	1	1	×		-	-				
	(SCK10)	Input	PIOR2 = 1	×	0	1	×		-	-				
		Output		0/1		0	1		-	-				
(SCL10)	Output	PIOR2 = 1	0/1	0	0	1	-	-						

**Note** PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.

**Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (9/15)**

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note 1</sup>	Alternate Function Output		80-pin	64-pin				
	Function Name	I/O							SAU Output Function	Other than SAU						
P45	P41	Input	-	-	0	1	×	0	-	-	√	√				
		Output				0	0/1									
	IVREF0	Analog input				1	1						×	1		
P46	P46	Input	-	×	-	1	×	0	(SCK00/SCL00) = 1	-	√	×				
		Output	×	0		0	0/1									
		N-ch open drain output		1												
	(SCK00) Input	PIOR1 = 1 <sup>Note 2</sup>		×									1	×		
	(SCK00) Output		0/1	0		1										
		(SCL00) Output	PIOR1 = 1 <sup>Note 2</sup>	0/1		0	1									
	SEG26	Output	×	0		0	0						1	-		
P47	P47	Input	-	×	-	1	×	0	(SDA00) = 1	-	√	×				
		Output	×	0		0	0/1									
		N-ch open drain output		1												
	(SI00) Input	PIOR1 = 1 <sup>Note 2</sup>		×									1	×		
	(RxD0) Input		PIOR1 = 1 <sup>Note 2</sup>	×		1	×									
	(SDA00) I/O		PIOR1 = 1 <sup>Note 2</sup>	1		0	1									
	SEG27	Output	×	0		0	0						1	-		
P50	P50	Input	-	-	-	1	×	0	-	-	√	×				
		Output				0	0/1									
	SEG4	Output				0	0						1			
P51	P51	Input	-	-	-	1	×	0	-	-	√	×				
		Output				1	0/1									
	SEG5	Output				0	0						1			
P52	P52	Input	-	-	-	1	×	0	-	-	√	√				
		Output				×	0						0/1			
	INTP1	Input				-	×						1	×		
	TI00	Input				PIOR0 = 0	×						1	×		
	TO00	Output				PIOR0 = 0	0						0	0		
	SEG6	Output				×	0						0	0	1	-

- Notes**
1. PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.
  2. 80-pin products only

**Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (10/15)**

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note</sup>	Alternate Function Output		80-pin	64-pin	
	Function Name	I/O							SAU Output Function	Other than SAU			
P53	P53	Input	-	×	-	1	×	0	-	-	√	√	
		Output		0		0	0/1						
		N-ch open drain output		1		0	0						
	INTP2	Input		×		1	×	1					
	SEG7	Output		0		0	0						
P54	P54	Input	-	×	-	1	×	0	-	-	√	√	
		Output	×	0		0	0/1						TO02 = 0
		N-ch open drain output	1	0		0							
	TI02	Input	PIOR0 = 0	×		1	×	-					
	TO02	Output	PIOR0 = 0	0		0	0						-
	SEG8	Output	×	0		0	0	1					-
P55	P55	Input	-	×	-	1	×	0	-	-	√	×	
		Output	-	0		0	0/1						
		N-ch open drain output	1	0		0							
	INTP5	Input	PIOR4 = 0	×		1	×	1					
	SEG9	Output	×	0		0	0						
P56	P56	Input	-	×	-	1	×	0	-	-	√	×	
		Output	×	0		0	0/1						TO06 = 0
		N-ch open drain output	1	0		0							
	TI06	Input	PIOR0 = 0	×		1	×	-					
	TO06	Output	PIOR0 = 0	0		0	0						-
	SEG10	Output	×	0		0	0	1					-
P57	P57	Input	-	-	-	1	×	0	-	-	√	√	
		Output				1	0/1						
	INTP6	Input				1	×	1					
	SEG11	Output				0	0						0

**Note** PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.



**Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (11/15)**

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note</sup>	Alternate Function Output		80-pin	64-pin
	Function Name	I/O							SAU Output Function	Other than SAU		
P60	P60	Input	-	-	-	1	×	-	-	-	√	√
		N-ch open drain output (6-V tolerance)	×			0	0/1			SCLA0 = 0 (TO01) = 0		
	SCLA0	I/O	×			0	0			(TO01) = 0		
	(TI01)	Input	PIOR0 = 1			1	×			-		
	(TO01)	Output	PIOR0 = 1			0	0			SCLA0 = 0		
P61	P61	Input	-	-	-	1	×	-	-	-	√	√
		N-ch open drain output (6-V tolerance)	×			0	0/1			SDAA0 = 0 (TO02) = 0		
	SDAA0	I/O	×			0	0			(TO02) = 0		
	(TI02)	Input	PIOR0 = 1			1	×			-		
	(TO02)	Output	PIOR0 = 1			0	0			SDAA0 = 0		
P70	P70	Input	-	-	-	1	×	0	-	-	√	√
		Output				0	0/1					
	KR0	Input				1	×					
	SEG12	Output				0	0					
P71	P71	Input	-	-	-	1	×	0	-	-	√	×
		Output				0	0/1					
	KR1	Input				1	×					
	SEG13	Output				0	0					
P72	P72	Input	-	-	-	1	×	0	-	-	√	×
		Output				0	0/1					
	KR2	Input				1	×					
	SEG14	Output				0	0					
P73	P73	Input	-	-	-	1	×	0	-	-	√	×
		Output				0	0/1					
	KR3	Input				1	×					
	SEG15	Output				0	0					

**Note** PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.

**Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (12/15)**

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note</sup>	Alternate Function Output		80-pin	64-pin
	Function Name	I/O							SAU Output Function	Other than SAU		
P74	P74	Input	-	-	-	1	×	0	-	-	√	√
		Output				0	0/1			TKBO00 = 0		
	KR4	Input				1	×			-		
	TKBO00	Output				0	0			-		
	SEG16	Output				0	0			1		
P75	P75	Input	-	-	-	1	×	0	-	-	√	√
		Output				0	0/1			TKBO01-2 = 0		
	KR5	Input				1	×			-		
	TKBO01-2	Output				0	0			-		
	SEG17	Output				0	0			1		
P76	P76	Input	-	-	-	1	×	0	-	-	√	√
		Output				0	0/1			TKBO01-1 = 0		
	KR6	Input				1	×			-		
	TKBO01-1	Output				0	0			-		
	SEG18	Output				0	0			1		
P77	P77	Input	-	-	-	1	×	0	-	-	√	√
		Output				0	0/1			TKBO01-0 = 0		
	KR7	Input				1	×			-		
	TKBO01-0	Output				0	0			-		
	SEG19	Output				0	0			1		

**Note** PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.

**Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (13/15)**

Pin Name	Used Function		CMC (EXCLK, OSCSEL, EXCLKS, OSCSELS)	Pxx	80-pin	64-pin
	Function Name	I/O				
P121	P121	Input	00xx/10xx/11xx	×	√	√
	X1	–	01xx	–		
P122	P122	Input	00xx/10xx/11xx	×	√	√
	X2	–	01xx	–		
	EXCLK	Input	11xx	–		
P123	P123	Input	xx00/xx10/xx11	×	√	√
	XT1	–	xx01	–		
P124	P124	Input	xx00/xx10/xx11	×	√	√
	XT2	–	xx01	–		
	EXCLKS	Input	xx11	–		

**Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (14/15)**

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note 1</sup>	Alternate Function Output		80-pin	64-pin
	Function Name	I/O							SAU Output Function	Other than SAU		
P125	P125	Input	–	–	–	1	×	1	–	–	√	√
		Output	×			0	0/1			(TO06) = 0		
	TI06 <sup>Note 3</sup>	Input	PIOR0 = 0			1	×			–		
	TO06 <sup>Note 3</sup>	Output	PIOR0 = 0			0	0			–		
	(TI06) <sup>Note 2</sup>	Input	PIOR0 = 1			1	×			–		
	(TO06) <sup>Note 2</sup>	Output	PIOR0 = 1			0	0			–		
	VL3	–	×			1	×			0		
P126	P126	Input	–	–	–	1	×	1	–	–	√	√
		Output	×			0	0/1			(TO04) = 0		
	(TI04)	Input	PIOR0 = 1			1	×			–		
	(TO04)	Output	PIOR0 = 1			0	0			–		
	CAPL	–	×			1	×			0		

- Notes**
1. PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.
  2. 80-pin products only
  3. 64-pin products only

**Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (15/15)**

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PFSEGxx (PFDEG, ISCVL3, ISCCAP) <sup>Note 1</sup>	Alternate Function Output		80-pin	64-pin	
	Function Name	I/O							SAU Output Function	Other than SAU			
P127	P127	Input	-	-	-	1	×	1	-	-	√	√	
		Output	×			0	0/1			(TO03/REMOOUT) = 0			
	(TI03)	Input	PIOR0 = 1			1	×			-			-
	(TO03)	Output	PIOR0 = 1			0	0			-			-
	(REMOOUT)	Output	PIOR0 = 1			-	-			-			-
	CAPH	-	×			1	×			0			-
P130	P30	Input	-	×	1	×	0	-	(SO00/TxD0) = 1	-	√	×	
		Output	×	0	0	0/1							
		N-ch open drain output	-	1	-	-							
	(SO00)	Output	PIOR1 = 1 <sup>Note 2</sup>	0	1	-							-
	(TxD0)	Output	PIOR1 = 1 <sup>Note 2</sup>	0	1	-							-
	SEG28	Output	×	0	0	0							0
P137	P137	Input	-	-	-	-	×	-	-	-	√	√	
	INTP0	Input	-	-	-	-	-	-	-	-	-	-	

- Notes**
1. PFDEG, ISCVL3, and ISCCAP are registers that correspond to P45, P125, and P126 and P127, respectively.
  2. 80-pin products only

**4.5.4 Operation of ports that alternately function as SEGxx pins**

The functions of ports that also serve as segment output pins (SEGxx) can be selected by using the port mode control register (PMCxx), port mode register (PMxx), and LCD port function registers 0 to 6 (PFSEG0 to PFSEG6).

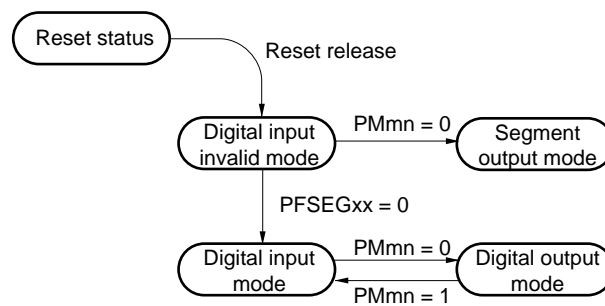
- (1) P00 to P07, P14 to P17, P30 to P35, P46, P47, P50 to P57, P70 to P77, P130 (ports that do not serve as analog input pins (ANLxx, IVREF0))

**Table 4-7. Settings of SEGxx/Port Pin Function**

PFSEGxx Bit of PFSEG0 to PFSEG6 Registers	PMxx Bit of PMxx Register	Pin Function	Initial Status
1	1	Digital input invalid mode	√
0	0	Digital output mode	–
0	1	Digital input mode	–
1	0	Segment output mode	–

The following shows the SEGxx/port pin function status transitions.

**Figure 4-12. SEGxx/Port Pin Function Status Transition Diagram**



**Caution** Be sure to set the segment output mode before segment output starts (while SCOC of LCD mode register 1 (LCDM1) is 0).

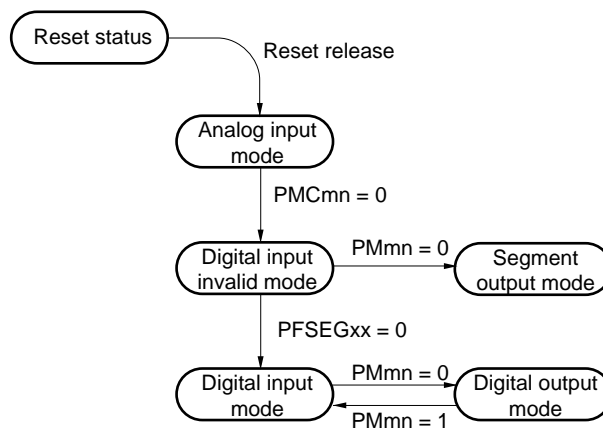
(2) P10 to P13, P22 to P27 (ports that serve as analog input pins (ANLxx))

Table 4-8. Settings of ANLxx/SEGxx/Port Pin Function

PMCxx Bit of PMCxx Register	PFSEGxx Bit of PFSEG3, PFSEG4 Registers	PMxx Bit of PMxx Register	Pin Function	Initial Status
1	1	1	Analog input mode	√
0	0	0	Digital output mode	–
0	0	1	Digital input mode	–
0	1	0	Segment output mode	–
0	1	1	Digital input invalid mode	–
Other than above			Setting prohibited	

The following shows the ANLxx/SEGxx/port pin function status transitions.

Figure 4-13. ANLxx/SEGxx/Port Pin Function Status Transition Diagram



**Caution** Be sure to set the segment output mode before segment output starts (while SCOC of LCD mode register 1 (LCDM1) is 0).

**4.5.5 Operation of ports that alternately function as VL3, CAPL, and CAPH pins**

The functions of the VL3/P125, CAPL/P126, and CAPH/P127 pins can be selected by using the LCD input switch control register (ISCLCD), LCD mode register 0 (LCDM0), and port mode register 12 (PM12).

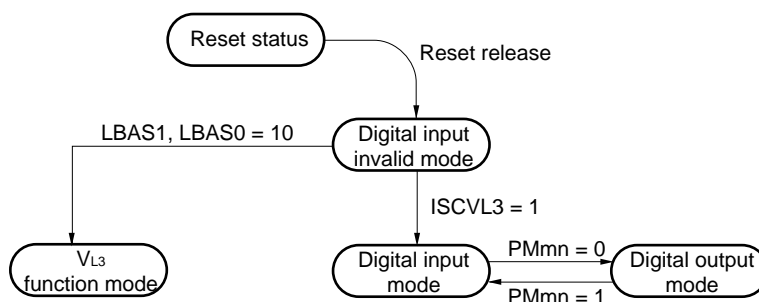
**(1) VL3/P125**

**Table 4-9. Settings of VL3/P125 Pin Function**

Bias Setting (LBAS1 and LBAS0 Bits of LCDM0 Register )	ISCVL3 Bit of ISCLCD Register	PM125 Bit of PM12 Register	Pin Function	Initial Status
Other than 1/4 bias method (LBAS1, LBAS0 = 00 or 01)	0	1	Digital input invalid mode	√
	1	0	Digital output mode	–
	1	1	Digital input mode	–
1/4 bias method (LBAS1, LBAS0 = 10)	0	1	VL3 function mode	–
Other than above			Setting prohibited	

The following shows the VL3/P125 pin function status transitions.

**Figure 4-14. VL3/P125 Pin Function Status Transition Diagram**



**Caution** Be sure to set the VL3 function mode before segment output starts (while SCOC of LCD mode register 1 (LCDM1) is 0).

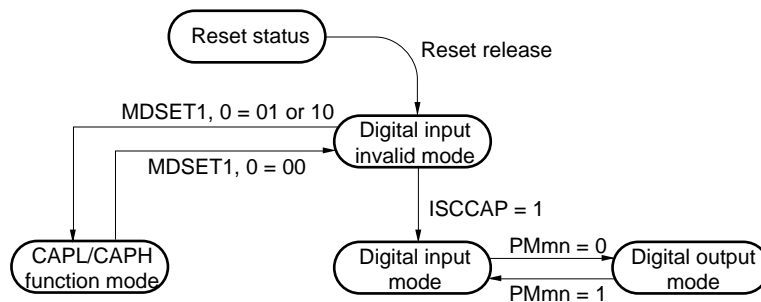
(2) CAPL/P126, CAPH/P127

Table 4-10. Settings of CAPL/P126, CAPH/P127 Pin Function

LCD Drive Voltage Generator (MDSET1 and MDSET0 Bits of LCDM0 Register)	ISCCAP Bit of ISCLCD Register	PM126, PM127 Bits of PM12 Register	Pin Function	Initial Status
External resistance division (MDSET1, MDSET0 = 00)	0	1	Digital input invalid mode	√
	1	0	Digital output mode	–
	1	1	Digital input mode	–
Internal voltage boosting or capacitor split (MDSET1, MDSET0 = 01 or 10)	0	1	CAPL/CAPH function mode	–
Other than above			Setting prohibited	

The following shows the CAPL/P126 and CAPH/P127 pin function status transitions.

Figure 4-15. CAPL/P126 and CAPH/P127 Pin Function Status Transition Diagram



**Caution** Be sure to set the CAPL/CAPH function mode before segment output starts (while SCOC of LCD mode register 1 (LCDM1) is 0).



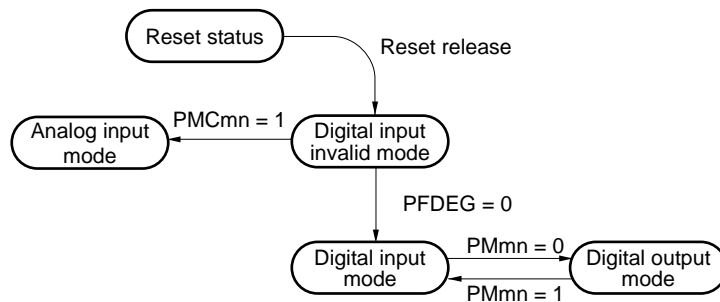
4.5.6 P45 (port that serves as an analog input pin (IVREF0))

Table 4-11. Settings of IVREF0/Port Pin Function

PMC45 Bit of PMC4 Register	PFDEG Bit of PFSEG3 Register	PM45 Bit of PM4 Register	Pin Function	Initial Status
1	1	1	Analog input mode	–
0	0	0	Digital output mode	–
0	0	1	Digital input mode	–
0	1	1	Digital input invalid mode	√
Other than above			Setting prohibited	

The following shows the IVREF0/port pin function status transitions.

Figure 4-16. IVREF0/Port Pin Function Status Transition Diagram



### 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 P10 is an output port, P11 to P17 are input ports (all pin statuses are high level), and the port latch value of port 1 is 00H, if the output of output port P10 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 1 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/L13.

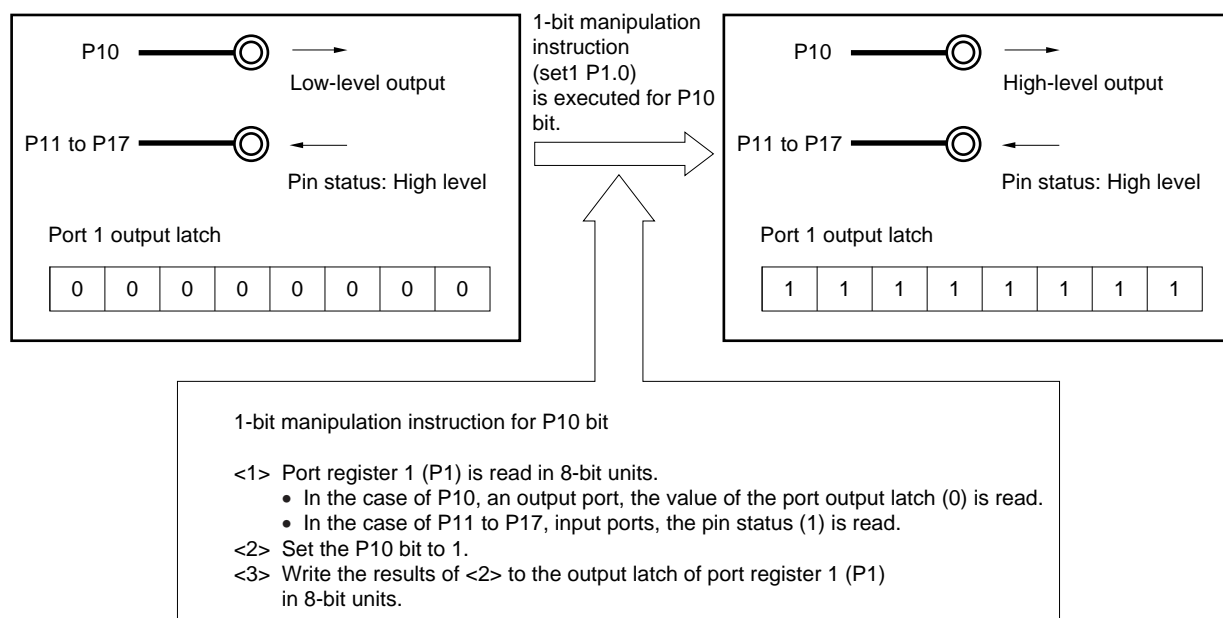
- <1> The Pn register is read in 8-bit units.
- <2> The targeted one bit is manipulated.
- <3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P17, which are input ports, are read. If the pin statuses of P11 to P17 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 4-17. Bit Manipulation Instruction (P10)



#### 4.6.2 Notes on specifying the pin settings

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

No specific setting is required for input pins because the output function 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 to lower power consumption.

## CHAPTER 5 CLOCK GENERATOR

## 5.1 Functions of Clock Generator

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

## (1) Main system clock

## &lt;1&gt; X1 oscillator

This circuit oscillates the X1 oscillator clock ( $f_x = 1$  to 20 MHz) by connecting a resonator to the X1 and X2 pins.

Oscillation can be stopped by executing the STOP instruction or setting the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

## &lt;2&gt; High-speed on-chip oscillator

The oscillation frequency ( $f_{HOCO}$ ) can be selected from 48, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz (typ.) by using the option byte (000C2H). When 48 MHz is selected as  $f_{HOCO}$ ,  $f_{IH}$  is set to 24 MHz. When 24 MHz or less is selected as  $f_{HOCO}$ ,  $f_{IH}$  is not divided and is set to the same frequency as  $f_{HOCO}$ . After reset release, the CPU always starts operating on this high-speed on-chip oscillator clock<sup>Note</sup>. 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 the option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 5-10 Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV)**.

The frequencies that can be specified for the high-speed on-chip oscillator by using the option byte and the high-speed on-chip oscillator frequency select register (HOCODIV) are shown below.

Power Supply Voltage	Oscillation Frequency (MHz)									
	1	2	3	4	6	8	12	16	24	48
$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	√	√	√	√	√	√	√	√	√	√
$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	√	√	√	√	√	√	√	√	–	–
$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	√	√	√	√	√	√	–	–	–	–
$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	√	√	√	√	–	–	–	–	–	–

An external main system clock ( $f_{EX} = 1$  to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. The external main system clock input can be disabled by executing the STOP instruction or setting 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 can be selected by setting 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 (VDD). The operating voltage of the flash memory must be set by using the CMODE0 and CMODE1 bits of the option byte (000C2H) (see CHAPTER 27 OPTION BYTE).

**Note** When 48 MHz is selected by setting the FRQSEL4 bit of the option byte (000C2H) to 1, the 48 MHz clock ( $f_{HOCO}$ ) is supplied to 16-bit timer KB20 and the 24 MHz clock, which is obtained by dividing  $f_{HOCO}$  by 2, is supplied to the other functions including the CPU. To supply the 48 MHz clock to 16-bit timer KB20, select  $f_{IH}$  as  $f_{CLK}$ .

&lt;R&gt;

**(2) Subsystem clock**

- **XT1 clock oscillator**

This circuit oscillates the XT1 oscillator clock ( $f_{XT} = 32.768$  kHz) by connecting a 32.768 kHz resonator to the 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**

This circuit oscillates the low-speed on-chip oscillator clock ( $f_{IL} = 15$  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 2
- 12-bit interval timer
- LCD controller/driver

This clock operates when either bit 4 (WDTON) of the option byte (000C0H) or 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 count clock of real-time clock 2 when the fixed-cycle interrupt function is used.

**Remark**

- fx: X1 clock oscillation frequency
- $f_{HOCO}$ : High-speed on-chip oscillator clock frequency (48 MHz max.)
- $f_{IH}$ : High-speed on-chip oscillator clock frequency (24 MHz max.)<sup>Note</sup>
- $f_{EX}$ : 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

**Note** The  $f_{IH}$  frequency can be controlled by hardware so that it becomes half the  $f_{HOCO}$  frequency when  $f_{HOCO}$  is set to 48 MHz by setting the FRQSEL4 bit of the option byte (000C2H) to 1, or the same frequency as  $f_{HOCO}$  when  $f_{HOCO}$  is set to 24 MHz or less by setting the FRQSEL4 bit of the option byte (000C2H) to 0.

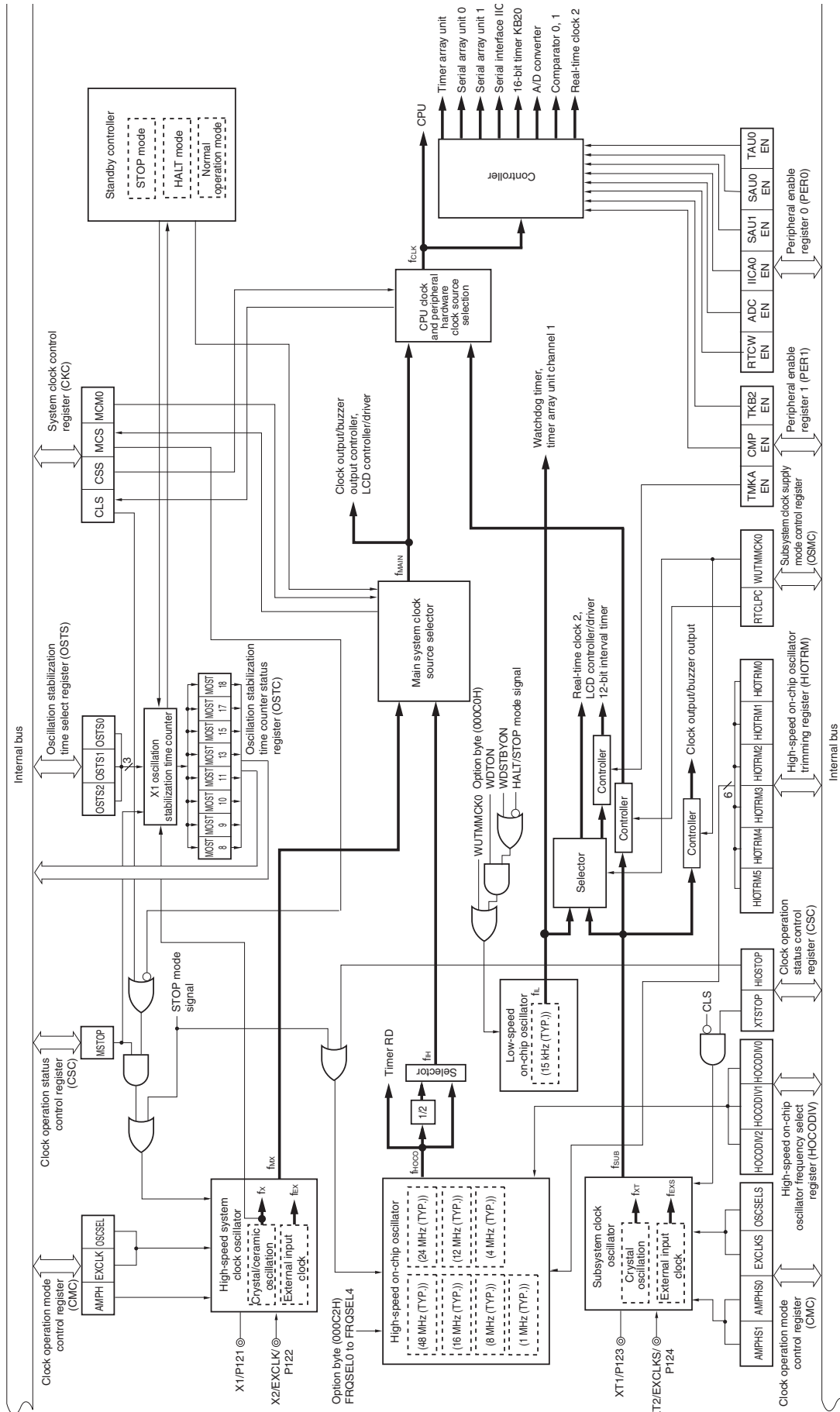
## 5.2 Configuration of Clock Generator

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 registers 0 and 1 (PER0, PER1) Subsystem clock supply mode control register (OSMC) High-speed on-chip oscillator frequency select register (HOCODIV) High-speed on-chip oscillator trimming register (HIOTRM)
Oscillators	X1 oscillator XT1 oscillator High-speed on-chip oscillator Low-speed on-chip oscillator

Figure 5-1. Block Diagram of Clock Generator



(Remark is listed on the next page.)

**Remark**

fx:	X1 clock oscillation frequency
f <sub>HOCO</sub> :	High-speed on-chip oscillator clock frequency (48 MHz max.)
f <sub>IH</sub> :	High-speed on-chip oscillator clock frequency (24 MHz max.) <sup>Note 1</sup>
f <sub>EX</sub> :	External main system clock frequency
f <sub>MX</sub> :	High-speed system clock frequency
f <sub>MAIN</sub> :	Main system clock frequency
f <sub>XT</sub> :	XT1 clock oscillation frequency
f <sub>EXS</sub> :	External subsystem clock frequency
f <sub>SUB</sub> :	Subsystem clock frequency <sup>Note 2</sup>
f <sub>CLK</sub> :	CPU/peripheral hardware clock frequency
f <sub>L</sub> :	Low-speed on-chip oscillator clock frequency

- Notes**
- The f<sub>IH</sub> frequency can be controlled by hardware so that it becomes half the f<sub>HOCO</sub> frequency when f<sub>HOCO</sub> is set to 48 MHz by setting the FRQSEL4 bit of the option byte (000C2H) to 1, or the same frequency as f<sub>HOCO</sub> when f<sub>HOCO</sub> is set to 24 MHz or less by setting the FRQSEL4 bit of the option byte (000C2H) to 0. To supply the 48 MHz clock to 16-bit timer KB20, select f<sub>IH</sub> as f<sub>CLK</sub>.
  - Selecting f<sub>SUB</sub> as the output clock of the clock output/buzzer output controller is prohibited when the WUTMMCK0 bit is set to 1.

### 5.3 Registers Controlling Clock Generator

The following 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 registers 0 and 1 (PER0, PER1)
- Subsystem clock supply mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- High-speed on-chip oscillator trimming register (HIOTRM)

**Caution** Which registers and bits are included depends on the product. Be sure to set registers and bits that are not mounted in a product to their initial values.

#### 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.

**Caution** The EXCLKS, OSCSELS, AMPHS1, and AMPHS0 bits are reset only by a power-on reset; they retain the previous values when a reset caused by another factor occurs.



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

Address: FFFA0H After reset: 00H<sup>Note</sup> R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS <sup>Note</sup>	OSCSELS <sup>Note</sup>	0	AMPHS1 <sup>Note</sup>	AMPHS0 <sup>Note</sup>	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 <sub>x</sub> ≤ 10 MHz						
	1	10 MHz < f <sub>x</sub> ≤ 20 MHz						

**Note** The EXCLKS, OSCSELS, AMPHS1, and AMPHS0 bits are reset only by a power-on reset; they retain the values when a reset caused by another factor occurs.

- Cautions 1.** The CMC register can be written only once after a reset ends, 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. A malfunction caused by mistakenly writing a value other than 00H is unrecoverable.
2. After a reset ends, set up the CMC register before setting the clock operation status control register (CSC) to start X1 or XT1 oscillation .
  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<sub>IH</sub> is selected as f<sub>CLK</sub> after a reset ends (before f<sub>CLK</sub> is switched to f<sub>MX</sub>).
  5. Count the f<sub>XT</sub> oscillation stabilization time by using software.

(The **cautions** continue and **Remark** is given on the next page.)

- Cautions 6.** Although the maximum system clock frequency is 24 MHz, the maximum frequency of the X1 oscillator is 20 MHz.
7. If a reset other than a power-on reset occurs after the CMC register is written and then the reset ends, be sure to set the CMC register to the value specified before the reset occurred, to prevent a malfunction if a program loop occurs.
  8. 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<sub>SS</sub> 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<sup>Note 1</sup>

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 <sub>CLK</sub> )
0	Main system clock (f <sub>MAIN</sub> )
1	Subsystem clock (f <sub>SUB</sub> )

CSS	Selection of CPU/peripheral hardware clock (f <sub>CLK</sub> )
0	Main system clock (f <sub>MAIN</sub> )
1 <sup>Note 2</sup>	Subsystem clock (f <sub>SUB</sub> )

MCS	Status of main system clock (f <sub>MAIN</sub> )
0	High-speed on-chip oscillator clock (f <sub>IH</sub> )
1	High-speed system clock (f <sub>MX</sub> )

MCM0 <sup>Note 2</sup>	Main system clock (f <sub>MAIN</sub> ) operation control
0	Selects the high-speed on-chip oscillator clock (f <sub>IH</sub> ) as the main system clock (f <sub>MAIN</sub> )
1	Selects the high-speed system clock (f <sub>MX</sub> ) as the main system clock (f <sub>MAIN</sub> )

**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.

**Remark** f<sub>HOCO</sub>: High-speed on-chip oscillator clock frequency (48 MHz max.)  
 f<sub>IH</sub>: High-speed on-chip oscillator clock frequency (24 MHz max.)<sup>Note</sup>  
 f<sub>MX</sub>: High-speed system clock frequency  
 f<sub>MAIN</sub>: Main system clock frequency  
 f<sub>SUB</sub>: Subsystem clock frequency

**Note** The f<sub>IH</sub> frequency can be controlled by hardware so that it becomes half the f<sub>HOCO</sub> frequency when f<sub>HOCO</sub> is set to 48 MHz by setting the FRQSEL4 bit of the option byte (000C2H) to 1, or the same frequency as f<sub>HOCO</sub> when f<sub>HOCO</sub> is set to 24 MHz or less by setting the FRQSEL4 bit of the option byte (000C2H) to 0. To supply the 48 MHz clock to 16-bit timer KB20, select f<sub>IH</sub> as f<sub>CLK</sub>.

(Cautions are listed on the next page.)

- Cautions**
1. Be sure to set bits 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 2, 12-bit interval timer, clock output/buzzer output controller, LCD controller/driver, and watchdog timer) is also changed at the same time. Consequently, you should 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 32 ELECTRICAL SPECIFICATIONS ( $T_A = -40$  to  $+85^\circ\text{C}$ ) and CHAPTER 33 ELECTRICAL SPECIFICATIONS ( $T_A = -40$  to  $+105^\circ\text{C}$ ).
  4. When selecting  $f_{HOCO}$  as the count source clock for 16-bit timer KB20, select  $f_{IH}$  as  $f_{CLK}$  before setting bit 4 (TKB2EN) of peripheral enable register 1 (PER1). When changing  $f_{CLK}$  to a clock other than  $f_{IH}$ , first clear bit 4 (TKB2EN) of peripheral enable register 1 (PER1).

### 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.

**Caution** The XTSTOP bit is reset only by a power-on reset; it retains the value when a reset caused by another factor occurs.

**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 <sup>Note</sup>	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	

**Note** The XTSTOP bit is reset only by a power-on reset; it retains the value when a reset caused by another factor occurs.

- Cautions**
1. After reset release, set the clock operation mode control register (CMC) before setting the CSC register.
  2. Set up the oscillation stabilization time select register (OSTS) before setting the MSTOP bit to 0 after releasing reset. Note that if the OSTS register is used with its default settings, setting the OSTS register is not required here.
  3. When starting X1 oscillation by setting 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<sub>CLK</sub>) by using the OSC register.

(The cautions continue on the next page.)

&lt;R&gt;

**Caution 6.** The setting of the flags of the register to stop clock oscillation (disabling the external clock input) and the condition before clock oscillation is stopped are shown in Table 5-2. Before stopping the clock oscillation, check the precondition for stopping the clock oscillation.

Table 5-2. Stopping the Clock

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

- If the X1 clock starts oscillating while the high-speed on-chip oscillator clock or subsystem clock is used as the CPU clock
- If the STOP mode is entered and then exited while the high-speed on-chip oscillator clock is used as the CPU clock and the X1 clock is oscillating

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

Occurrence of a reset signal, executing the STOP instruction, or setting MSTOP (bit 7 of clock operation status control register (CSC)) to 1 clears 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 exited

**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	$2^8/f_x$ max.	25.6 $\mu$ s max.	12.8 $\mu$ s max.
1	0	0	0	0	0	0	0	$2^9/f_x$ min.	25.6 $\mu$ s min.	12.8 $\mu$ s min.
1	1	0	0	0	0	0	0	$2^9/f_x$ min.	51.2 $\mu$ s min.	25.6 $\mu$ s min.
1	1	1	0	0	0	0	0	$2^{10}/f_x$ min.	102 $\mu$ s min.	51.2 $\mu$ s min.
1	1	1	1	0	0	0	0	$2^{11}/f_x$ min.	204 $\mu$ s min.	102 $\mu$ s min.
1	1	1	1	1	0	0	0	$2^{13}/f_x$ min.	819 $\mu$ s min.	409 $\mu$ s min.
1	1	1	1	1	1	0	0	$2^{15}/f_x$ min.	3.27 ms min.	1.63 ms min.
1	1	1	1	1	1	1	0	$2^{17}/f_x$ min.	13.1 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	$2^{18}/f_x$ min.	26.2 ms min.	13.1 ms min.

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

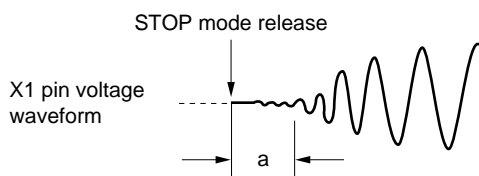
**2.** The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS).

In the following cases, set the oscillation stabilization time of the OSTS register to a value greater than the count value to be monitored by using the OSTC register after the oscillation starts.

- To start X1 clock oscillation while the high-speed on-chip oscillator clock or subsystem clock is used as the CPU clock.
- To enter and exit the STOP mode while the high-speed on-chip oscillator clock is used as the CPU clock and the X1 clock is 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 exited.)

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

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

Reset signal generation sets the OSTS register to 07H.



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

Address: FFFA3H After reset: 07H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection	Oscillation stabilization time selection	
				$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$
0	0	0	$2^8/f_x$	25.6 $\mu\text{s}$	12.8 $\mu\text{s}$
0	0	1	$2^9/f_x$	51.2 $\mu\text{s}$	25.6 $\mu\text{s}$
0	1	0	$2^{10}/f_x$	102 $\mu\text{s}$	51.2 $\mu\text{s}$
0	1	1	$2^{11}/f_x$	204 $\mu\text{s}$	102 $\mu\text{s}$
1	0	0	$2^{13}/f_x$	819 $\mu\text{s}$	409 $\mu\text{s}$
1	0	1	$2^{15}/f_x$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_x$	13.1 ms	6.55 ms
1	1	1	$2^{18}/f_x$	26.2 ms	13.1 ms

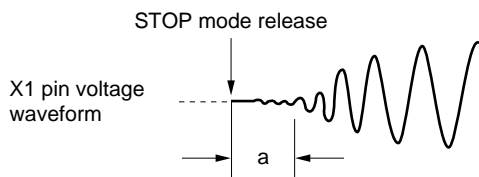
**Cautions 1.** Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

**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 a value greater than the count value to be monitored by using the OSTC register after the oscillation starts.

- To start X1 clock oscillation while the high-speed on-chip oscillator clock or subsystem clock is used as the CPU clock.
- To enter and exit the STOP mode while the high-speed on-chip oscillator clock is used as the CPU clock and the X1 clock is 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 exited.)

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



**Remark**  $f_x$ : X1 clock oscillation frequency

### 5.3.6 Peripheral enable registers 0 and 1 (PER0, PER1)

These registers are used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware not used is also stopped so as to reduce the power consumption and noise.

To use the peripheral functions below, which are controlled by these registers, set the bit corresponding to each function to 1 before initial setup of the peripheral functions.

- Real-time clock 2
- A/D converter
- Serial interface IICA0
- Serial array unit 1
- Serial array unit 0
- Timer array unit
- 12-bit interval timer
- Comparators 0 and 1
- 16-bit timer KB20

The PER0 and PER1 registers can be set by a 1-bit or 8-bit memory manipulation instruction. Reset signal generation clears these registers 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	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

RTCWEN	Control of real-time clock 2 (RTC2) input clock supply
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFRs used by the real-time clock 2 (RTC2) cannot be written.</li> <li>• The real-time clock 2 (RTC2) is operable.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>• SFRs used by the real-time clock 2 (RTC2) can be read and written.</li> <li>• The real-time clock 2 (RTC2) is operable.</li> </ul>

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFRs used by the A/D converter cannot be written.</li> <li>• The A/D converter is in the reset status.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>• SFRs used by the A/D converter can be read and written.</li> </ul>

**Caution** Be sure to clear bits 6 and 1 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	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

IICA0EN	Control of serial interface IICA0 input clock supply
0	Stops input clock supply. <ul style="list-style-type: none"> <li>SFRs used by serial interface IICA0 cannot be written.</li> <li>Serial interface IICA0 is in the reset status.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>SFRs used by serial interface IICA0 can be read and written.</li> </ul>

SAU1EN	Control of serial array unit 1 input clock supply
0	Stops input clock supply. <ul style="list-style-type: none"> <li>SFRs used by serial array unit 1 cannot be written.</li> <li>Serial array unit 1 is in the reset status.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>SFRs used by serial array unit 1 can be read and written.</li> </ul>

SAU0EN	Control of serial array unit 0 input clock supply
0	Stops input clock supply. <ul style="list-style-type: none"> <li>SFRs used by serial array unit 0 cannot be written.</li> <li>Serial array unit 0 is in the reset status.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>SFRs used by serial array unit 0 can be read and written.</li> </ul>

TAU0EN	Control of timer array unit input clock supply
0	Stops input clock supply. <ul style="list-style-type: none"> <li>SFRs used by timer array unit cannot be written.</li> <li>Timer array unit is in the reset status.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>SFRs used by timer array unit can be read and written.</li> </ul>

**Caution** Be sure to clear bits 6 and 1 to "0".

Figure 5-8. Format of Peripheral Enable Register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
PER1	TMKAEN	0	CMPEN	TKB2EN	0	0	0	0

TMKAEN	Control of 12-bit interval timer input clock supply
0	Stops input clock supply. <ul style="list-style-type: none"> <li>SFRs used by the 12-bit interval timer cannot be written.</li> <li>The 12-bit interval timer is in the reset status.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>SFRs used by the 12-bit interval timer can be read and written.</li> </ul>

CMPEN	Control of comparators 0/1 input clock supply
0	Stops input clock supply. <ul style="list-style-type: none"> <li>SFRs used by comparators 0 and 1 cannot be written.</li> <li>Comparators 0 and 1 are in the reset status.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>SFRs used by comparators 0 and 1 can be read and written.</li> </ul>

TKB2EN <sup>Note</sup>	Control of 16-bit timer KB20 input clock supply
0	Stops input clock supply. <ul style="list-style-type: none"> <li>SFRs used by 16-bit timer KB20 cannot be written.</li> <li>16-bit timer KB20 is in the reset status.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>SFRs used by 16-bit timer KB20 can be read and written.</li> </ul>

**Note** When FRQSEL4 in the user option byte (000C2H) is 1, select  $f_{IH}$  as  $f_{CLK}$  before setting bit 4 (TKB2EN) of peripheral enable register 1 (PER1). When changing  $f_{CLK}$  to a clock other than  $f_{IH}$ , first clear bit 4 (TKB2EN) of peripheral enable register 1 (PER1).

**Caution** Be sure to clear bits 6 and 3 to 0 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 other than real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver is stopped in STOP mode or in HALT mode while the subsystem clock is selected as the CPU clock.

In addition, the OSMC register can be used to select the operation clock of real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver.

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

Reset signal generation clears this register to 00H.

**Figure 5-9. 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 in HALT mode while subsystem clock is selected as CPU clock
0	Enables supplying the subsystem clock to peripheral functions (See <b>Tables 21-1 to 21-3</b> for peripheral functions whose operations are enabled.)
1	Stops supplying the subsystem clock to peripheral functions other than real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver.

WUTMMCK0	Selection of operation clock for real-time clock 2, 12-bit interval timer, and LCD controller/driver	Selection of clock output from PCLBUZn pin of clock output/buzzer output controller
0	Subsystem clock ( $f_{SUB}$ )	Selecting the subsystem clock ( $f_{SUB}$ ) is enabled.
1	Low-speed on-chip oscillator clock ( $f_{IL}$ )	Selecting the subsystem clock ( $f_{SUB}$ ) is disabled.

- Cautions**
1. Be sure to select the subsystem clock (WUTMMCK0 bit = 0) if the subsystem clock is oscillating.
  2. When WUTMMCK0 is set to 1, the low-speed on-chip oscillator clock oscillates.
  3. When WUTMMCK0 is set to 1, only the constant-period interrupt function of real-time clock 2 can be used. The year, month, day of the week, day, hour, minute, and second counters and the 1 Hz output function of real-time clock 2 cannot be used. The interval of the constant-period interrupt is calculated by constant period (value selected by using the RTCC0 register)  $\times f_{SUB}/f_{IL}$ .
  4. The subsystem clock and low-speed on-chip oscillator clock can only be switched by using the WUTMMCK0 bit if real-time clock 2, 12-bit interval timer, and LCD controller/driver are all stopped.
  5. Do not select  $f_{SUB}$  as the clock output or buzzer output clock when the WUTMMCK0 bit is 1.

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

This register is used to change the high-speed on-chip oscillator frequency set by an option byte (000C2H). However, the selectable frequency depends on the FRQSEL4 and FRQSEL3 bits of the option byte (000C2H).

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 5-10. Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV)**

Address: F00A8H After reset: undefined R/W

Symbol	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		
			FRQSEL4 = 0		FRQSEL4 = 1
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0
0	0	0	f <sub>IH</sub> = 24 MHz	Setting prohibited	f <sub>IH</sub> = 24 MHz f <sub>HOCO</sub> = 48 MHz
0	0	1	f <sub>IH</sub> = 12 MHz	f <sub>IH</sub> = 16 MHz	f <sub>IH</sub> = 12 MHz f <sub>HOCO</sub> = 24 MHz
0	1	0	f <sub>IH</sub> = 6 MHz	f <sub>IH</sub> = 8 MHz	f <sub>IH</sub> = 6 MHz f <sub>HOCO</sub> = 12 MHz
0	1	1	f <sub>IH</sub> = 3 MHz	f <sub>IH</sub> = 4 MHz	f <sub>IH</sub> = 3 MHz f <sub>HOCO</sub> = 6 MHz
1	0	0	Setting prohibited	f <sub>IH</sub> = 2 MHz	Setting prohibited
1	0	1	Setting prohibited	f <sub>IH</sub> = 1 MHz	Setting prohibited
Other than above			Setting prohibited		

**Cautions 1. For the HOCODIV register, specify a value in the operating voltage range corresponding to the flash operation mode specified in the option byte (000C2H), regardless of whether the frequency is changed.**

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE2			
0	0	LV (low-voltage main) mode	1 to 4 MHz	1.6 to 5.5 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 5.5 V
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		

2. **Specify the HOCODIV register settings after first selecting the high-speed on-chip oscillator clock (f<sub>IH</sub>) as the CPU/peripheral hardware clock (f<sub>CLK</sub>).**
3. **After changing the frequency setting by using the HOCODIV register, the system will begin operating on the new frequency after the transition time shown below has elapsed.**
  - **Operation for up to three clocks at the pre-change frequency**
  - **CPU/peripheral hardware clock wait at the post-change frequency for up to three clocks**

### 5.3.9 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<sub>DD</sub> pin voltage change after accuracy adjustment. When the temperature and V<sub>DD</sub> voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.

Figure 5-11. Format of High-Speed On-Chip Oscillator Trimming Register (HIOTRM)

Address: F00A0H After reset: Undefined<sup>Note</sup> 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	

**Note** The value after reset is the value adjusted at shipment.

- Remarks**
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, refer to the application note for **RL78 MCU Series High-speed On-chip Oscillator (HOCO) Clock Frequency Correction (R01AN0464)**.

## 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.

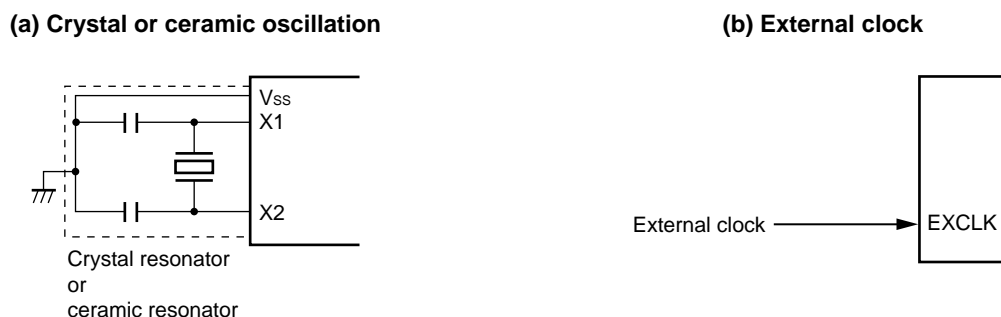
- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

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

When the X1 and X2 pins are not used as input port pins, either, see **Table 2-2 Connection of Unused Pins**.

Figure 5-12 shows an example of the external circuit connected to the X1 oscillator.

**Figure 5-12. Example of External Circuit Connected to X1 Oscillator**



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

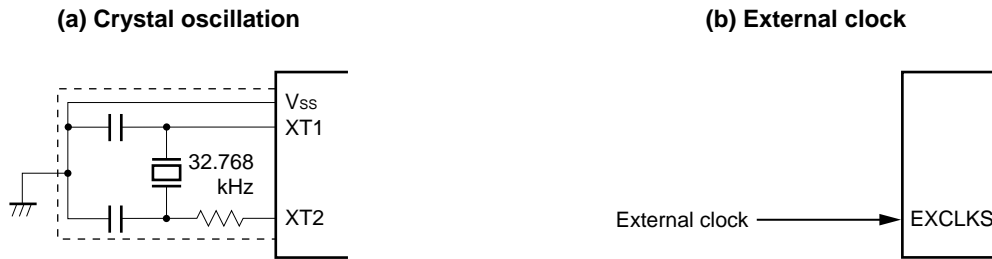
When the XT1 oscillator is not used, specify the input port mode (EXCLKS, OSCSELS = 0, 0).

When the XT1 and XT2 pins are not used as input port pins, either, see **Table 2-2 Connection of Unused Pins**.

Figure 5-13 shows an example of the external circuit connected to the XT1 oscillator.



Figure 5-13. Example of External Circuit Connected to XT1 Oscillator



**Caution** When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-12 and 5-13 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<sub>SS</sub>. Do not ground the capacitor to a ground pattern through which a high current flows.
- Do not fetch signals from the oscillator.

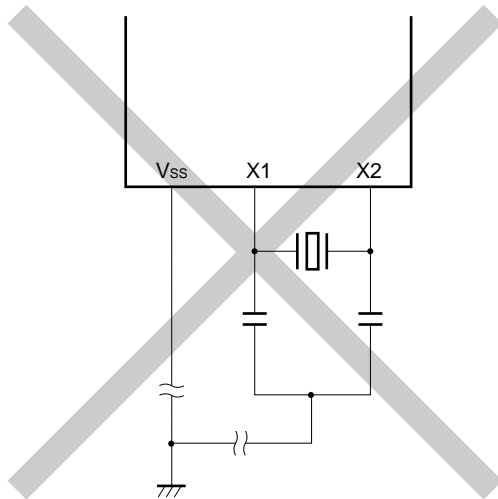
The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- 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<sub>SS</sub> 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.

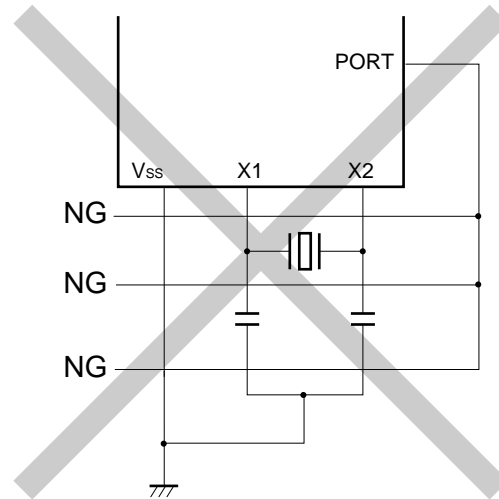
Figure 5-14 shows examples of incorrect resonator connection.

**Figure 5-14. Examples of Incorrect Resonator Connection (1/2)**

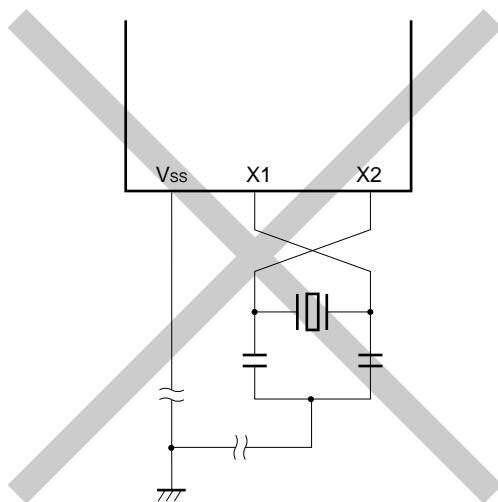
**(a) Too long wiring**



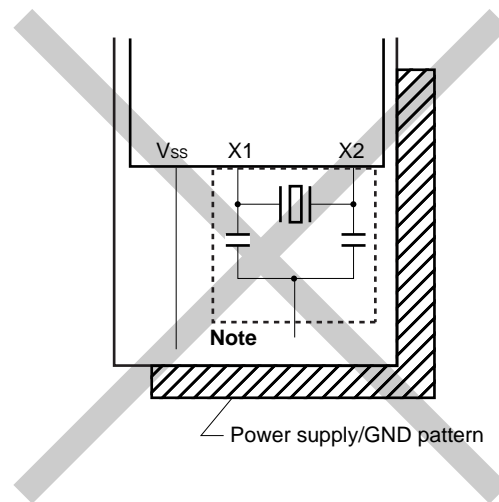
**(b) Crossed signal line**



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



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

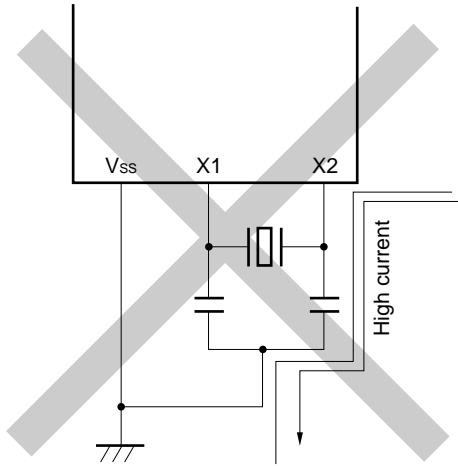


**Note** Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board. Do not configure a layout that will cause capacitance elements and affect the oscillation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

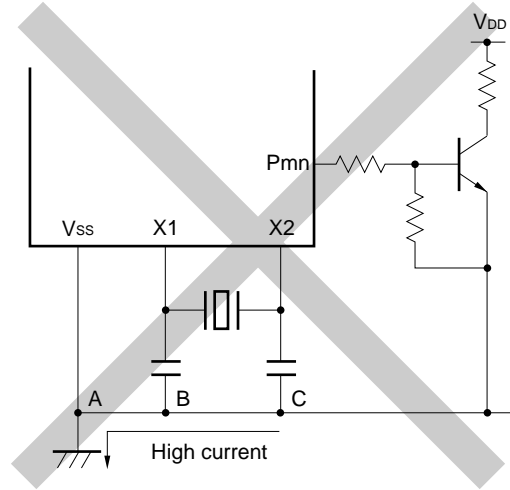
**Remark** When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 5-14. Examples of Incorrect Resonator Connection (2/2)

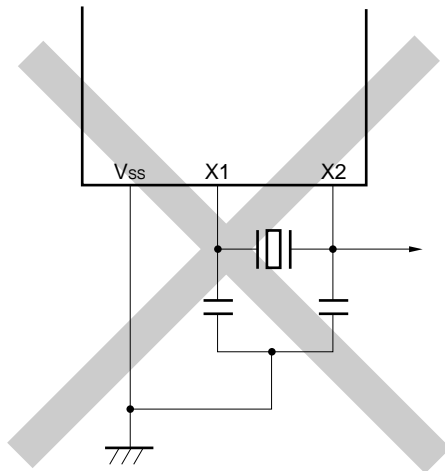
(e) Wiring near high alternating current



(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

A high-speed on-chip oscillator is incorporated in the RL78/L13. The frequency can be selected from among 48, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using the option byte (000C2H). When 48 MHz is selected, the oscillation clock is divided by 2 and supplied as the CPU clock. Oscillation can be controlled by using 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

A low-speed on-chip oscillator is incorporated in the RL78/L13.

The low-speed on-chip oscillator clock is used only as the clock for the watchdog timer, real-time clock 2, 12-bit interval timer, and the LCD controller/driver. The low-speed on-chip oscillator clock cannot be used as the CPU clock.

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

<R> As long as the watchdog timer is not operating and WUTMMCK0 is not zero, the low-speed on-chip oscillator continues oscillating. 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. The low-speed on-chip oscillator clock does not stop even if a program loop that stops the system occurs while the watchdog timer is operating.

## 5.5 Clock Generator Operation

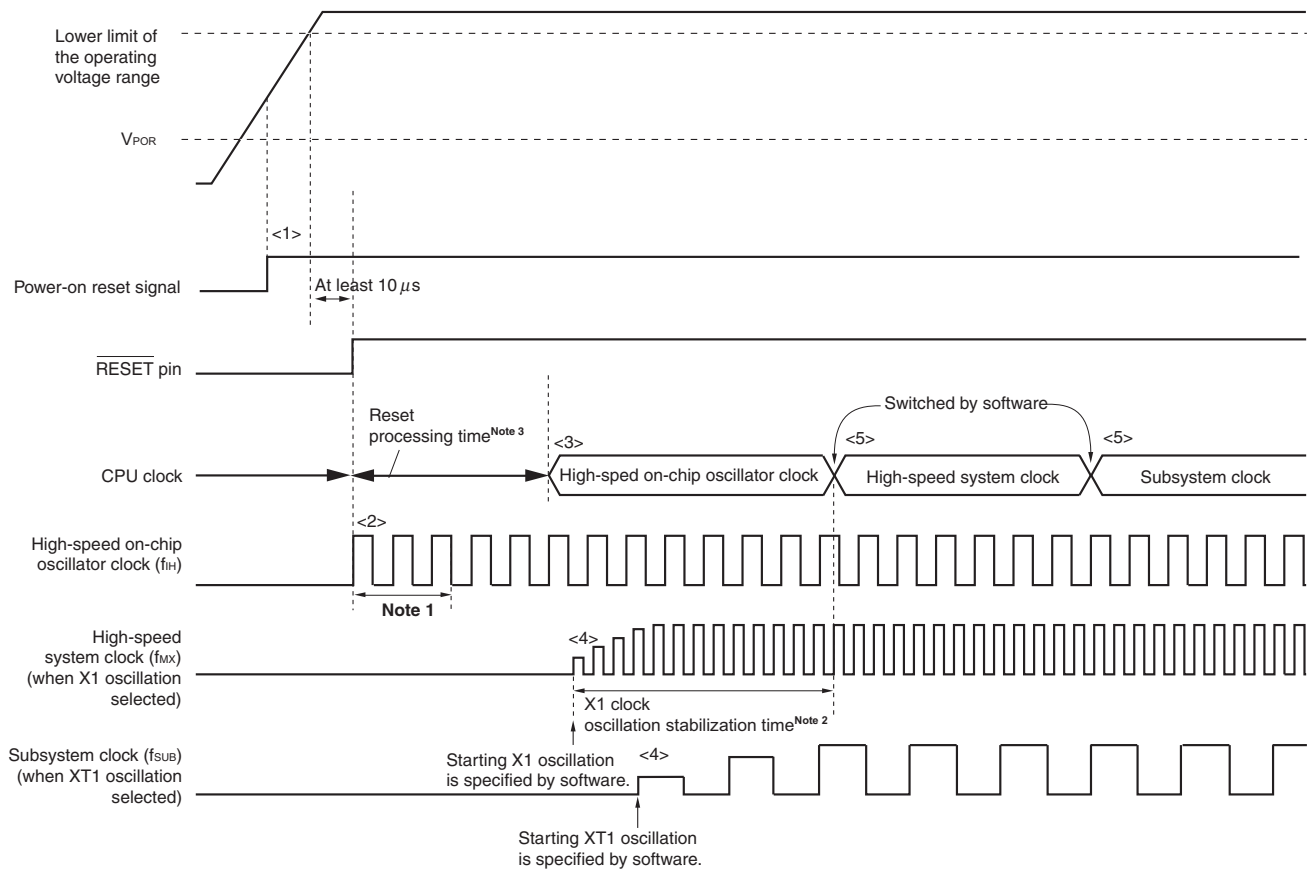
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 5-1**).

- Main system clock  $f_{\text{MAIN}}$ 
  - High-speed system clock  $f_{\text{MX}}$ 
    - X1 clock  $f_{\text{X}}$
    - External main system clock  $f_{\text{EX}}$
  - High-speed on-chip oscillator clock  $f_{\text{IH}}$
- Subsystem clock  $f_{\text{SUB}}$ 
  - XT1 clock  $f_{\text{XT}}$
  - External subsystem clock  $f_{\text{EXS}}$
- Low-speed on-chip oscillator clock  $f_{\text{IL}}$
- CPU/peripheral hardware clock  $f_{\text{CLK}}$

In the RL78/L13, the CPU starts operating when the high-speed on-chip oscillator starts generating the clock after reset release .

The clock generator operation after the power supply voltage is turned on is shown in Figure 5-15.

Figure 5-15. 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 **32.4 AC Characteristics** and **33.4 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 23 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 reset release, the high-speed on-chip oscillator clock is used as the CPU/peripheral hardware clock ( $f_{CLK}$ ). The frequency of the high-speed on-chip oscillator can be selected from 48, 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz by using FRQSEL0 to FRQSEL4 of the option byte (000C2H). The frequency can also be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

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

CMODE1	CMODE0	Setting of flash operation mode	
0	0	LV (low voltage main) mode	$V_{DD} = 1.6\text{ V to }5.5\text{ V @ }1\text{ MHz to }4\text{ MHz}$
1	0	LS (low speed main) mode	$V_{DD} = 1.8\text{ V to }5.5\text{ V @ }1\text{ MHz to }8\text{ MHz}$
1	1	HS (high speed main) mode	$V_{DD} = 2.4\text{ V to }5.5\text{ V @ }1\text{ MHz to }16\text{ MHz}$ $V_{DD} = 2.7\text{ V to }5.5\text{ V @ }1\text{ MHz to }24\text{ MHz}$
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
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	1 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		
			FRQSEL4 = 0		FRQSEL4 = 1
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0
0	0	0	$f_{IH} = 24$ MHz	Setting prohibited	$f_{IH} = 24$ MHz $f_{HOCO} = 48$ MHz
0	0	1	$f_{IH} = 12$ MHz	$f_{IH} = 16$ MHz	$f_{IH} = 12$ MHz $f_{HOCO} = 24$ MHz
0	1	0	$f_{IH} = 6$ MHz	$f_{IH} = 8$ MHz	$f_{IH} = 6$ MHz $f_{HOCO} = 12$ MHz
0	1	1	$f_{IH} = 3$ MHz	$f_{IH} = 4$ MHz	$f_{IH} = 3$ MHz $f_{HOCO} = 6$ MHz
1	0	0	Setting prohibited	$f_{IH} = 2$ MHz	Setting prohibited
1	0	1	Setting prohibited	$f_{IH} = 1$ MHz	Setting prohibited
Other than above			Setting prohibited		



**5.6.2 Example of setting X1 oscillation clock**

After reset release, the high-speed on-chip oscillator clock is used as the CPU/peripheral hardware clock ( $f_{CLK}$ ). To change the clock to the X1 oscillation clock, specify the oscillator settings by using the oscillation stabilization time select register (OSTS), clock operation mode control register (CMC), and clock operation status control register (CSC) to start oscillation, and then make sure that oscillation has stabilized by checking the oscillation stabilization time counter status register (OSTC). After the oscillation stabilizes, select the X1 oscillation clock as  $f_{CLK}$  by using the system clock control register (CKC).

[Register settings] Set the register according to steps <1> to <5> below.

<1> Set the OSCSEL bit of the CMC register to 1. If  $f_x$  is higher than 10 MHz, set the AMPH bit to 1, to start 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

<2> Using the OSTS register, select the oscillation stabilization time of the X1 oscillator after the STOP mode is exited.

Example: Specify as below to wait for oscillation to stabilize for at least 102.4  $\mu$ s when using 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 the MSTOP bit of the CSC register to 0 to start oscillation of 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 are set to the following values to wait for at least 102.4  $\mu$ s for oscillation to stabilize when using 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**
- 1 The EXCLKS, OSCSELS, AMPHS1, AMPHS0, and XTSTOP bits are reset only by a power-on reset; they retain the previous values when a reset caused by another factor occurs.
  - 2 Keep the operating voltage within the range that allows operation of the flash memory as set in an option byte (000C2H) before and after changes to the main system clock ( $f_{MAIN}$ ) by using the system clock control register (CKC).

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
0	0	LV (low-voltage main) mode	1 MHz to 4 MHz	1.6 V to 5.5 V
1	0	LS (low-speed main) mode	1 MHz to 8 MHz	1.8 V to 5.5 V
1	1	HS (high-speed main) mode	1 MHz to 16 MHz	2.4 V to 5.5 V
			1 MHz to 32 MHz	2.7 V to 5.5 V

### 5.6.3 Example of setting XT1 oscillation clock

After reset release, the high-speed on-chip oscillator clock is used as the CPU/peripheral hardware clock ( $f_{CLK}$ ). To change the clock to the XT1 oscillation clock, specify the oscillator settings by using the subsystem clock supply mode control register (OSMC), clock operation mode control register (CMC), and clock operation status control register (CSC) to start oscillation, and then select the XT1 oscillation clock as  $f_{CLK}$  by using the system clock control register (CKC).

[Register settings] Set the register according to steps <1> to <5> below.

<1> Set the RTCLPC bit to 1 to run only the real-time clock 2, 12-bit interval timer, and LCD controller/driver on the subsystem clock (for ultra-low current consumption) in the STOP mode or HALT mode during CPU operation on the subsystem clock.

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

<2> Set the OSCSELS bit of the CMC register to 1 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: Use these bits to specify the oscillation mode of the XT1 oscillator.

<3> Clear the XTSTOP bit of the CSC register to 0 to start oscillation of 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 features such as the timer 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

**Caution** The EXCLKS, OSCSELS, AMPHS1, AMPHS0, and XTSTOP bits are reset only by a power-on reset; they retain the previous values when a reset caused by another factor occurs.

5.6.4 CPU clock status transition diagram

Figure 5-16 shows the CPU clock status transition diagram of this product.

Figure 5-16. CPU Clock Status Transition

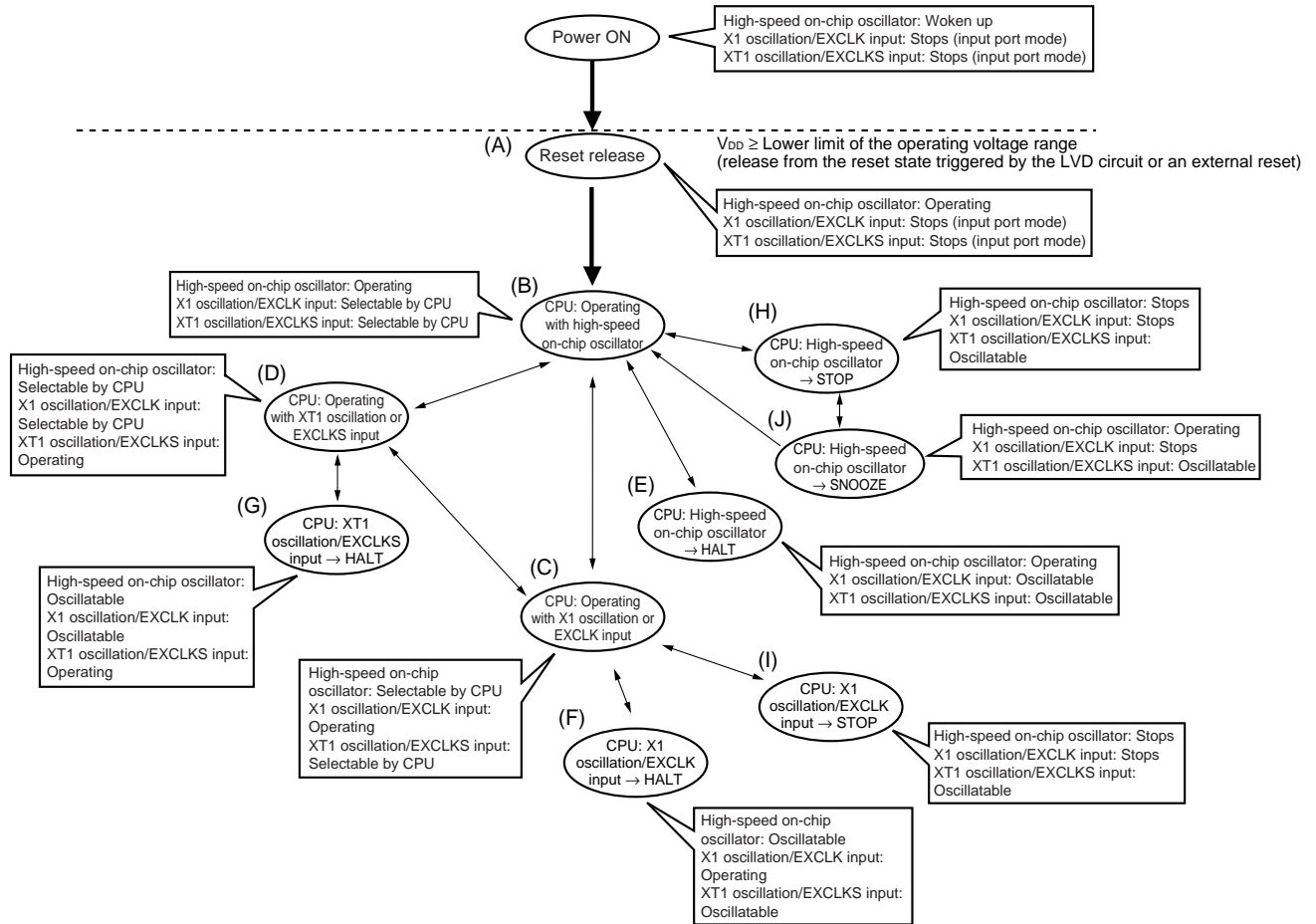


Table 5-3 shows transition of the CPU clock and examples of setting the special function registers (SFRs).

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

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

Status Transition	SFR Setting
(A) → (B)	SFR setting not required (SFRs are in the default status after reset release).

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

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

(SFR setting sequence) →

SFR Flag to Set Status Transition	CMC Register <sup>Note 1</sup>			OSTS Register	CSC Register MSTOP	OSTC Register	CKC Register MCM0
	EXCLK	OSCSEL	AMPH				
(A) → (B) → (C) (X1 clock: 1 MHz ≤ f <sub>x</sub> ≤ 10 MHz)	0	1	0	<b>Note 2</b>	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < f <sub>x</sub> ≤ 20 MHz)	0	1	1	<b>Note 2</b>	0	Must be checked	1
(A) → (B) → (C) (external main clock)	1	1	×	<b>Note 2</b>	0	Not need to be checked	1

- Notes**
- The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release.
  - Set the oscillation stabilization time as follows.
    - Desired oscillation stabilization time indicated by the oscillation stabilization time counter status register (OSTC) ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

**Caution** Specify the clock after the supply voltage has reached the operable voltage of the clock to be specified (see CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C) and CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)).

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

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

(SFR setting sequence) →

SFR Flag to Set Status Transition	CMC Register <sup>Note</sup>				CSC Register XTSTOP	Waiting for Oscillation Stabilization	CKC Register CSS
	EXCLKS	OSCSELS	AMPHS1	AMPHS0			
(A) → (B) → (D) (XT1 clock)	0	1	0/1	0/1	0	Necessary	1
(A) → (B) → (D) (external subsystem 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**
- ×: don't care
  - (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-16.

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

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

(SFR setting sequence) →

Status Transition	SFR Flag to Set			CMC Register <sup>Note 1</sup>			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP			MCM0		
(B) → (C) (X1 clock: 1 MHz ≤ f <sub>x</sub> ≤ 10 MHz)	0	1	0	<b>Note 2</b>	0	Must be checked	1			
(B) → (C) (X1 clock: 10 MHz < f <sub>x</sub> ≤ 20 MHz)	0	1	1	<b>Note 2</b>	0	Must be checked	1			
(B) → (C) (external main clock)	1	1	×	<b>Note 2</b>	0	Not need to be checked	1			

Setting unnecessary if these bits are already set
 Setting unnecessary if the CPU is operating on the high-speed system clock

- Notes**
- The clock operation mode control register (CMC) can be changed only once after reset release. This setting is not necessary if it has already been set.
  - Set the oscillation stabilization time as follows.
    - Desired oscillation stabilization time indicated by the oscillation stabilization time counter status register (OSTC) ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

**Caution** Specify the clock after the supply voltage has reached the operable voltage of the clock to be specified (see CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C) and CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)).

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

(Setting sequence of SFR registers) →

Status Transition	Setting Flag of SFR Register			CMC Register <sup>Note</sup>			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**
- x: don't care
  - (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-16.

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

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

(SFR setting sequence) →

SFR Flag to Set	CSC Register	Oscillation Accuracy Stabilization Time	CKC Register
	HIOSTOP		MCM0
Status Transition			
(C) → (B)	0	<b>Note</b>	0

Setting unnecessary if the CPU is operating on the high-speed on-chip oscillator clock

**Note** When FRQSEL4 = 0: 18 to 65 μs

<R> When FRQSEL4 = 1: 18 to 135 μs

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

(SFR setting sequence) →

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

Setting unnecessary if the CPU is operating on the subsystem clock

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

(Setting sequence of SFRs)

Setting Flag of SFR	CSC Register	Oscillation Accuracy Stabilization Time	CKC Register	
	HIOSTOP		CSS	MCM0
Status Transition				
(D) → (B)	0	<b>Note</b>	0	0

Setting unnecessary if the CPU is operating on the high-speed on-chip oscillator clock

Setting unnecessary if this bit is already set

**Note** When FRQSEL4 = 0: 18 to 65 μs

<R> When FRQSEL4 = 1: 18 to 135 μs

**Remarks 1.** (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-16.

**2.** The oscillation accuracy stabilization time changes according to the temperature conditions and the STOP mode period.

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

(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 <sub>x</sub> ≤ 10 MHz)	<b>Note</b>	0	Must be checked	0
(D) → (C) (X1 clock: 10 MHz < f <sub>x</sub> ≤ 20 MHz)	<b>Note</b>	0	Must be checked	0
(D) → (C) (external main clock)	<b>Note</b>	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 oscillation stabilization time indicated by the oscillation stabilization time counter status register (OSTC) ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

**Caution** Specify the clock after the supply voltage has reached the operable voltage of the clock to be specified (see CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C) and CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)).

(10) • HALT mode (E) entered while CPU is operating on high-speed on-chip oscillator clock (B)

- HALT mode (F) entered while CPU is operating on high-speed system clock (C)
- HALT mode (G) entered while CPU is operating on subsystem clock (D)

Status Transition	Setting
(B) → (E) (C) → (F) (D) → (G)	Execute HALT instruction

**Remark** (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-16.

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

- (11) • **STOP mode (H) entered while CPU is operating on high-speed on-chip oscillator clock (B)**
- **STOP mode (I) entered while CPU is operating on high-speed system clock (C)**

(Setting sequence) →

Status Transition		Setting		
(B) → (H)		Stopping peripheral functions that are disabled in STOP mode	–	Execute the STOP instruction.
(C) → (I)	X1 oscillation clock		Set up the OSTS register.	
	External main system clock		–	

**(12) Changing CPU operating mode from STOP mode (H) to SNOOZE mode (J)**

For details about the setting for switching from the STOP mode to the SNOOZE mode, see **12.8 SNOOZE Mode Function**, **14.5.7 SNOOZE mode function**, and **14.6.3 SNOOZE mode function**.

**Remark** (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-16.



5.6.5 Conditions before changing the CPU clock and processing after changing CPU clock

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

Table 5-4. Changing CPU Clock (1/2)

CPU Clock		Conditions Before Change	Processing After Change
Before Change	After Change		
<R> High-speed on-chip oscillator clock	X1 clock	X1 oscillation is stable • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed	The operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1) after checking that the CPU clock is changed.
	External main system clock	Inputting the external clock from the EXCLK pin is enabled • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	XT1 clock	XT1 oscillation is stable • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0	
	External subsystem clock	Inputting the external clock from the EXCLKS pin is enabled • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	External main system clock	Transition impossible	–
	XT1 clock	XT1 oscillation is stable • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	External subsystem clock	Inputting the external clock from the EXCLKS pin is enabled • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	Inputting the external main system clock can be disabled (MSTOP = 1).
	X1 clock	Transition impossible	–
	XT1 clock	XT1 oscillation is stable • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed	Inputting the external main system clock can be disabled (MSTOP = 1).
	External subsystem clock	Inputting the external clock from the EXCLKS pin is enabled • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	Inputting the external main system clock can be disabled (MSTOP = 1).

Table 5-4. Changing CPU Clock (2/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
XT1 clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is oscillating and the high-speed on-chip oscillator clock is selected as the main system clock • HIOSTOP = 0, MCS = 0	XT1 oscillation can be stopped (XTSTOP = 1)
	X1 clock	X1 oscillation is stable and the high-speed system clock is selected as the main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed • MCS = 1	
	External main system clock	Inputting the external clock from the EXCLK pin is enabled and the high-speed system clock is selected as the main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	External subsystem clock	Transition impossible	–
External subsystem clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is oscillating and the high-speed on-chip oscillator clock is selected as the main system clock • HIOSTOP = 0, MCS = 0	Inputting external subsystem clock can be disabled (XTSTOP = 1).
	X1 clock	X1 oscillation is stable and the high-speed system clock is selected as the main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed • MCS = 1	
	External main system clock	Inputting the external clock from the EXCLK pin is enabled and the high-speed system clock is selected as the main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	XT1 clock	Transition impossible	–

**5.6.6 Time required for switching 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 clock is not switched immediately after rewriting the CKC register; operation continues on the clock before the change for several clock cycles (see **Tables 5-5 to 5-7**).

Whether the CPU is operating on the main system clock or the subsystem clock can be checked by 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 checked by using bit 5 (MCS) of the CKC register.

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

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

Clock A	Switching Directions	Clock B	Remark
f <sub>IH</sub>	↔	f <sub>MX</sub>	See <b>Table 5-6</b> .
f <sub>MAIN</sub>	↔	f <sub>SUB</sub>	See <b>Table 5-7</b> .

**Table 5-6. Maximum Number of Clock Cycles Required for Switching Between f<sub>IH</sub> and f<sub>MX</sub>**

Value Before Switchover		Value After Switchover	
MCM0		MCM0	
		0 (f <sub>MAIN</sub> = f <sub>IH</sub> )	1 (f <sub>MAIN</sub> = f <sub>MX</sub> )
0 (f <sub>MAIN</sub> = f <sub>IH</sub> )	f <sub>MX</sub> ≥ f <sub>IH</sub>		2 clock cycles
	f <sub>MX</sub> < f <sub>IH</sub>		2 f <sub>IH</sub> /f <sub>MX</sub> clock cycles
1 (f <sub>MAIN</sub> = f <sub>MX</sub> )	f <sub>MX</sub> ≥ f <sub>IH</sub>	2 f <sub>MX</sub> /f <sub>IH</sub> clock cycles	
	f <sub>MX</sub> < f <sub>IH</sub>	2 clock cycles	

**Table 5-7. Maximum Number of Clocks Required for Switching Between f<sub>MAIN</sub> and f<sub>SUB</sub>**

Value Before Switchover		Value After Switchover	
CSS		CSS	
		0 (f <sub>CLK</sub> = f <sub>MAIN</sub> )	1 (f <sub>CLK</sub> = f <sub>SUB</sub> )
0 (f <sub>CLK</sub> = f <sub>MAIN</sub> )			1 + 2 f <sub>MAIN</sub> /f <sub>SUB</sub> clock cycles
1 (f <sub>CLK</sub> = f <sub>SUB</sub> )		3 clock cycles	

- Remarks 1.** The number of clock cycles in Table 5-6 and Table 5-7 is the number of CPU clock cycles before switchover.
- 2.** Calculate the number of clock cycles in Table 5-6 and Table 5-7, rounding off the decimal values.

**Example** When switching the main system clock from the high-speed system clock to the high-speed on-chip oscillator clock (when f<sub>IH</sub> = 8 MHz, f<sub>MX</sub> = 10 MHz)  
 2 f<sub>MX</sub>/f<sub>IH</sub> cycles = 2 (10/8) = 2.5 → 3 clock cycles

### 5.6.7 Conditions before stopping clock oscillation

<R> 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 5-8. Conditions Before Stopping the Clock Oscillation and Flag Settings**

Clock	Conditions Before Stopping Clock Oscillation (Disabling External Clock Input)	SFR Flag Settings
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 oscillator clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
External main system clock		
XT1 oscillator clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock.)	XTSTOP = 1
External subsystem clock		

## 5.7 Resonator and Oscillator Constants

<R> For the resonators for which the operation has been verified and their oscillation constants (for reference), see the page for the corresponding product at the Renesas Web site (<http://www.renesas.com>).

- 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 5-17. External Oscillation Circuit Example

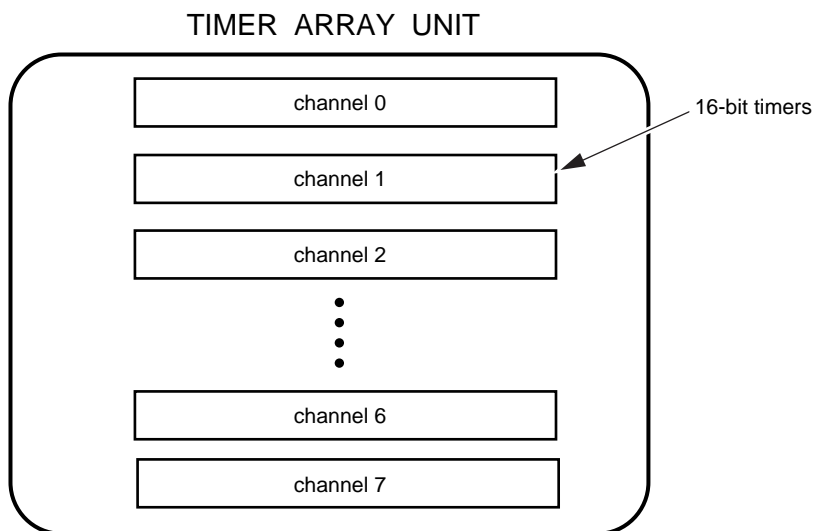


CHAPTER 6 TIMER ARRAY UNIT

- Cautions 1.** The presence or absence of timer I/O pins depends on the product. See Table 6-2 Timer I/O Pins provided in Each Product for details.
- 2.** Most of the following descriptions in this chapter use the 80-pin as an example.

The timer array unit has eight 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
<ul style="list-style-type: none"> <li>• Interval timer (→ see 6.8.1)</li> <li>• Square wave output (→ see 6.8.1)</li> <li>• External event counter (→ see 6.8.2)</li> <li>• Input pulse interval measurement (→ see 6.8.3)</li> <li>• Measurement of high-/low-level width of input signal (→ see 6.8.4)</li> <li>• Delay counter (→ see 6.8.5)</li> </ul>	<ul style="list-style-type: none"> <li>• One-shot pulse output (→ see 6.9.1)</li> <li>• PWM output (→ see 6.9.2)</li> <li>• Multiple PWM output (→ see 6.9.3)</li> <li>• Remote control output function (→ see 6.9.4)</li> </ul>

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

- Interval timer (higher 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)

Channel 7 can be used to realize LIN-bus communication operating in combination with UART0 of the serial array unit.

### 6.1 Functions of Timer Array Unit

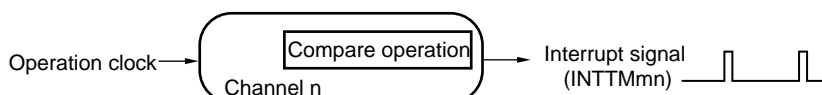
The timer array unit has the following functions.

#### 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.

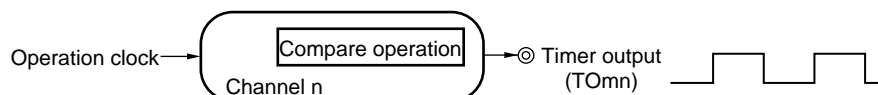
##### (1) Interval timer

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



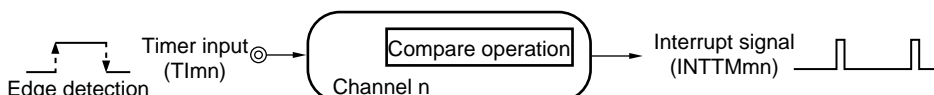
##### (2) Square wave output

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).



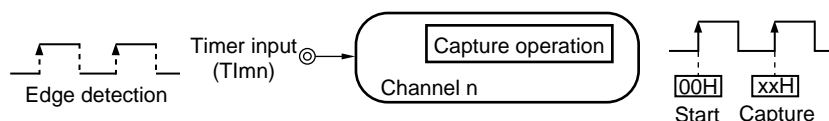
##### (3) External event counter

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) Input pulse interval measurement

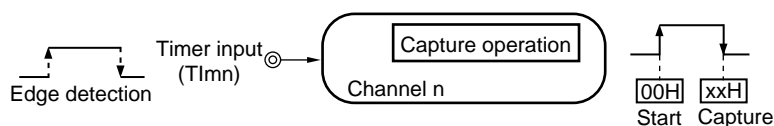
Counting is started by the valid edge of a pulse signal input to a timer input pin (TImn). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



(Remark is listed on the next page.)

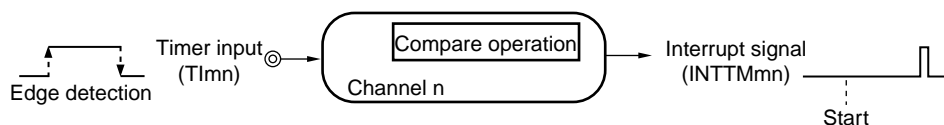
**(5) Measurement of high-/low-level width of input signal**

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.



**(6) 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.



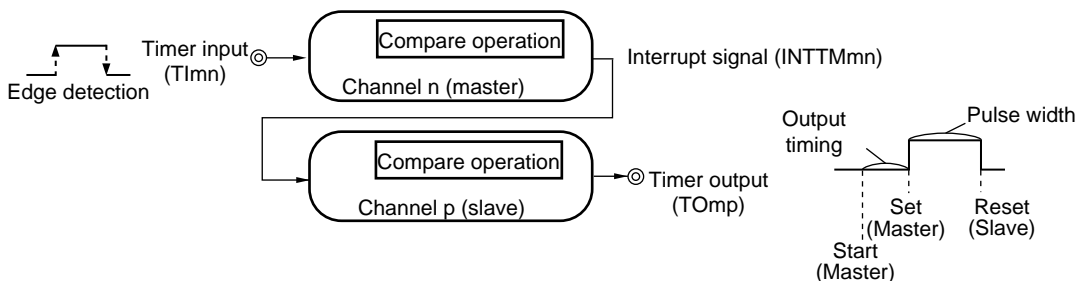
**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

**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.

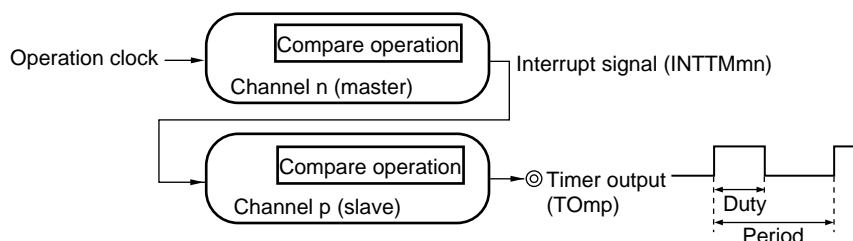
**(1) One-shot pulse output**

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.

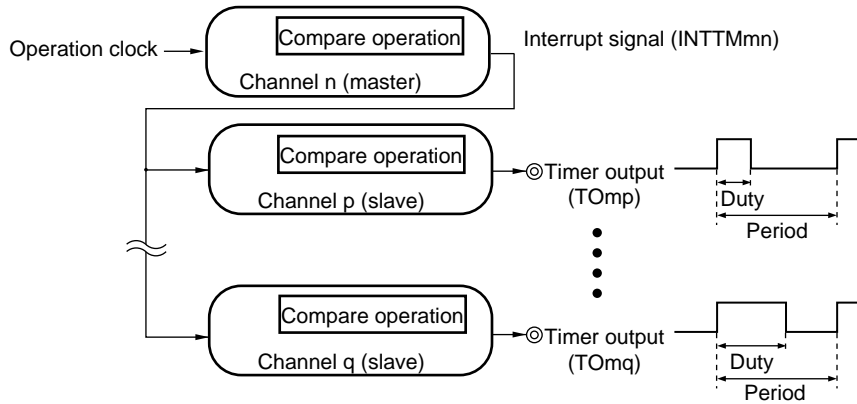


(Caution and Remark are listed on the next page.)



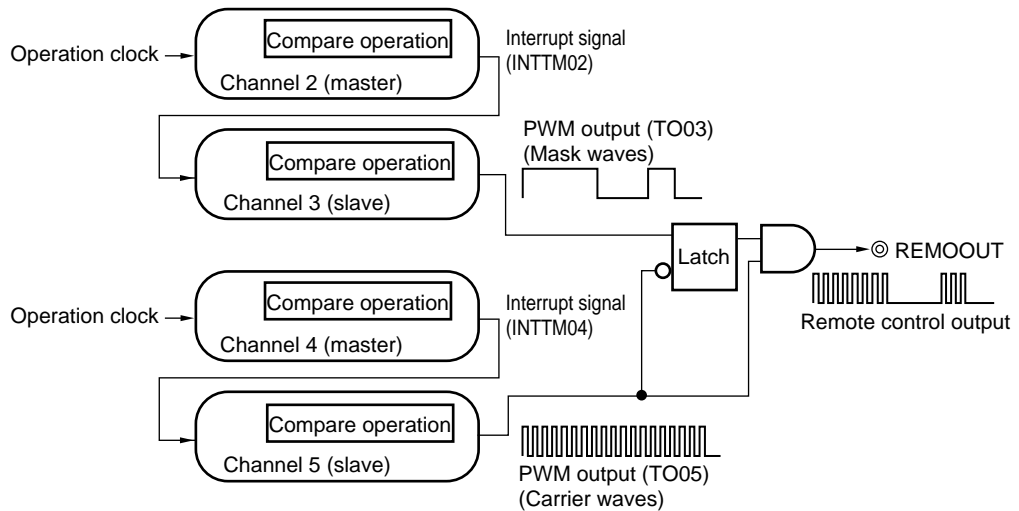
**(3) Multiple PWM (Pulse Width Modulation) output**

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



**(4) Remote control output function**

The pairings of channels 2 and 3 and channels 4 and 5 are used to output the PWM signal. The PWM signal output from channel 3 is used as a mask waves, the PWM signal output from channel 5 is used as a carrier waves, and the logical products of these signals are output as remote control output.



**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 7),  
p, q: Slave channel number (n < p < q ≤ 7)

**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).

#### 6.1.4 LIN-bus supporting function (channel 7 only)

Timer array unit is used to check whether signals received in LIN-bus communication match the LIN-bus communication format.

##### (1) Detection of wakeup signal

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD0) of UART0 and the count value of the timer is captured at the rising edge. In this way, a low-level width can be measured. If the low-level width is greater than a specific value, it is recognized as a wakeup signal.

##### (2) Detection of break field

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD0) of UART0 after a wakeup signal is detected, and the count value of the timer is captured at the rising edge. In this way, a low-level width is measured. If the low-level width is greater than a specific value, it is recognized as a break field.

##### (3) Measurement of pulse width of sync field

After a break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD0) of UART0 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

**Remark** For details about setting up the operations used to implement the LIN-bus, see **6.3.14 Input switch control register (ISC)** and **6.8.4 Operation as input signal high-/low-level width measurement**.

## 6.2 Configuration of Timer Array Unit

The 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 TI07 <sup>Note 1</sup> , RxD0 pin (for LIN-bus)
Timer output	TO00 to TO07 <sup>Note 1</sup> , output controller
Control registers	<p>&lt;Registers of unit setting block&gt;</p> <ul style="list-style-type: none"> <li>• Peripheral enable register 0 (PER0)</li> <li>• Timer clock select register m (TPSm)</li> <li>• Timer channel enable status register m (TEm)</li> <li>• Timer channel start register m (TSM)</li> <li>• Timer channel stop register m (TTm)</li> <li>• Timer input select register 0 (TIS0)</li> <li>• Timer output select register (TOS)</li> <li>• Timer output enable register m (TOEm)</li> <li>• Timer output register m (TOM)</li> <li>• Timer output level register m (TOLm)</li> <li>• Timer output mode register m (TOMm)</li> </ul> <p>&lt;Registers of each channel&gt;</p> <ul style="list-style-type: none"> <li>• Timer mode register mn (TMRmn)</li> <li>• Timer status register mn (TSRmn)</li> <li>• Input switch control register (ISC)</li> <li>• Noise filter enable register 1 (NFEN1)</li> <li>• Port mode control register (PMCxx)<sup>Note 2</sup></li> <li>• Port mode register (PMxx)<sup>Note 2</sup></li> <li>• Port register (Pxx)<sup>Note 2</sup></li> </ul>

**Notes** 1. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

2. The port mode control register (PMCxx), port mode registers (PMxx) and port registers (Pxx) to be set differ depending on the product. For details, see **6.3.16 Registers controlling port functions of pins to be used for timer I/O**.

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

The port pins alternatively used as timer I/O pins in each timer array unit channel depend on the product.

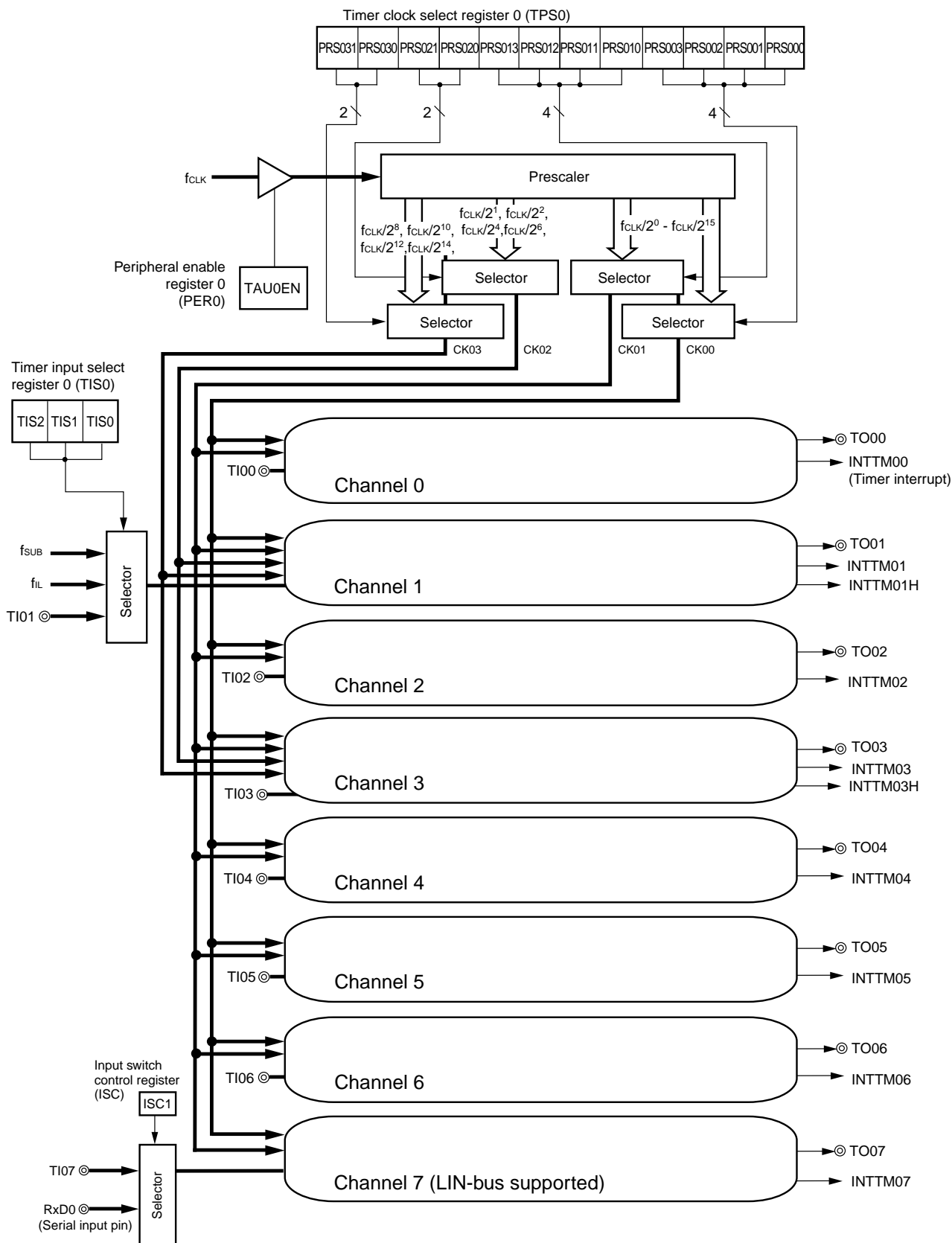
**Table 6-2. Timer I/O Pins Provided in Each Product**

Timer Array Unit Channels	80-pin	64-pin
Channel 0	P52/TI00/TO00 (P40)	
Channel 1	P32/TI01/TO01 (P60)	
Channel 2	P54/TI02/TO02 (P61)	
Channel 3	P30/TI03/TO03/REMOOUT (P127)	
Channel 4	P14/TI04/TO04 (P126)	
Channel 5	P42/TI05/TO05 (P01)	
Channel 6	P56/TI06/TO06 (P125)	P125/TI06/TO06
Channel 7	P15/TI07/TO07 (P41)	P15/TI07/TO07

- Remarks**
1. Because timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.
  2. Pins in the parentheses indicate an alternate port when the bit 0 of the peripheral I/O redirection register (PIOR) is set to "1".

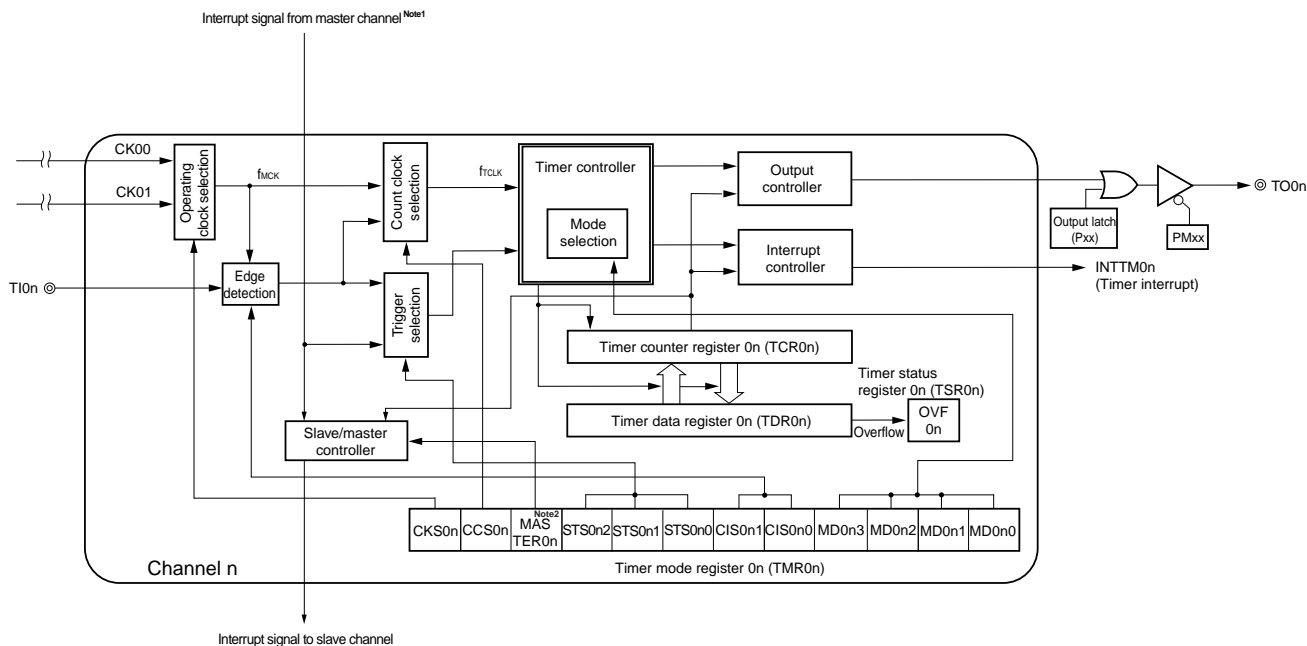
Figures 6-1 to 6-6 show block diagrams of the timer array unit.

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, 2, 4, 6 of Timer Array Unit



Notes 1. Channels 2, 4, and 6 only

2. n = 2, 4, 6 only

Remark n = 0, 2, 4, 6

Figure 6-3. Internal Block Diagram of Channels 1 of Timer Array Unit

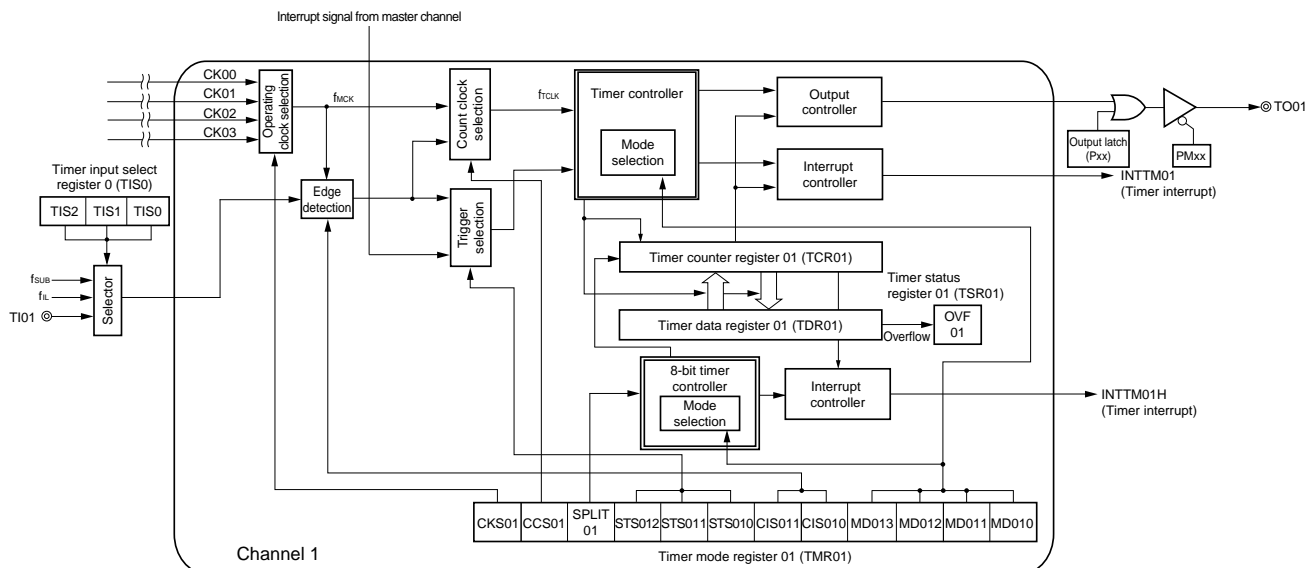


Figure 6-4. Internal Block Diagram of Channels 3 of Timer Array Unit

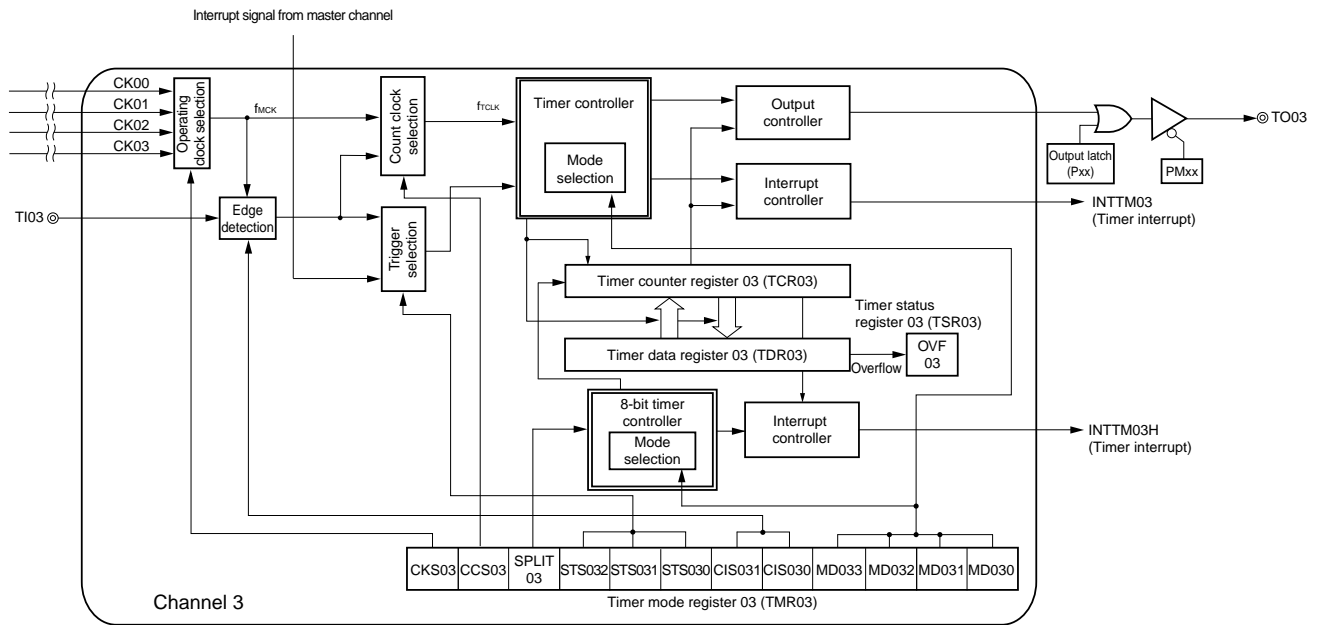


Figure 6-5. Internal Block Diagram of Channels 5 of Timer Array Unit

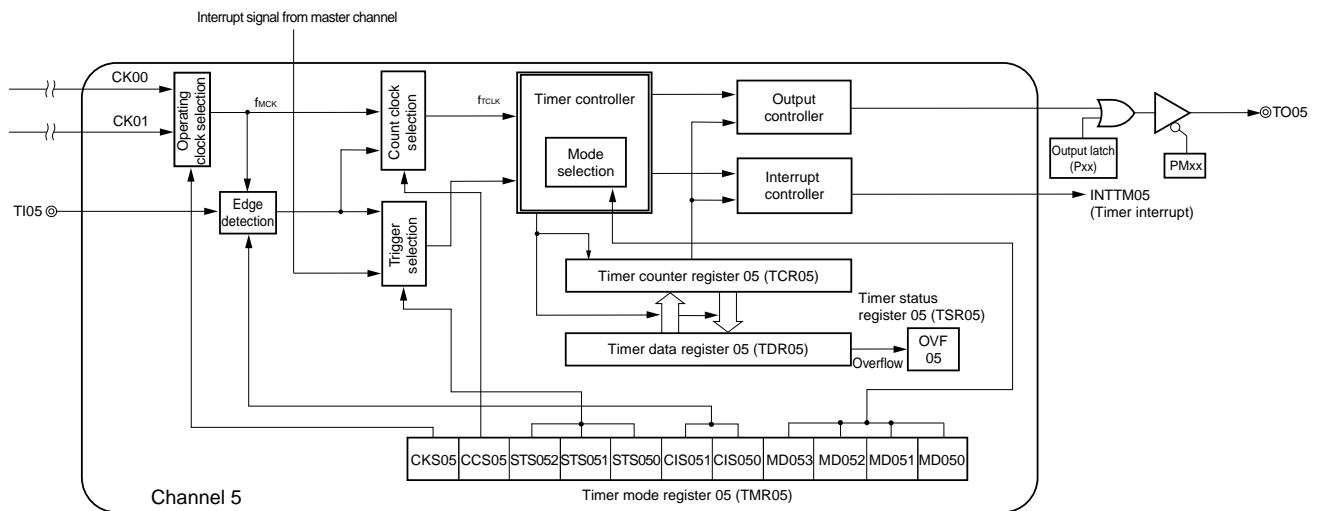
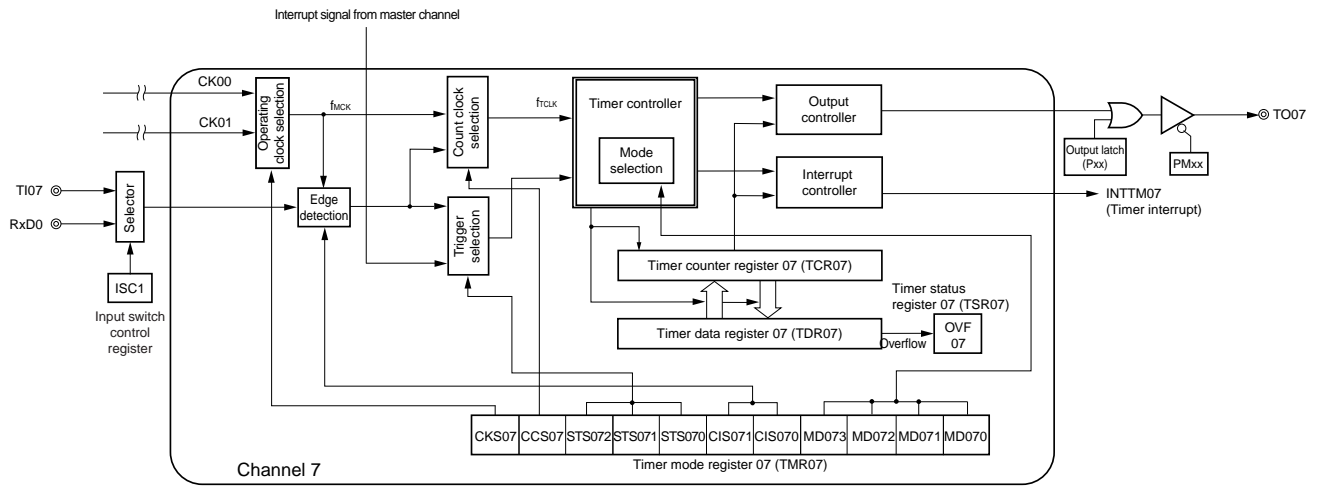


Figure 6-6. Internal Block Diagram of Channels 7 of Timer Array Unit





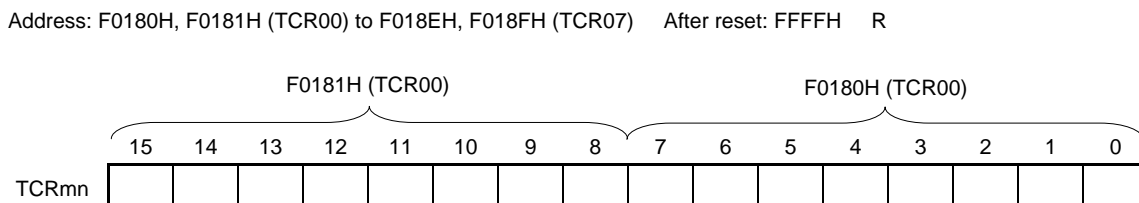
**6.2.1 Timer count register mn (TCRmn)**

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) (see **6.3.3 Timer mode register mn (TMRmn)**).

**Figure 6-7. Format of Timer Count Register mn (TCRmn)**



**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

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 TAUmEN 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 <sup>Note</sup>			
		Value if the operation mode was changed after releasing reset	Value if the Operation was restarted after count operation paused (TTmn = 1)	Value if the operation mode was changed after count operation paused (TTmn = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Count down	FFFFH	Value if stop	Undefined	–
Capture mode	Count up	0000H	Value if stop	Undefined	–
Event counter mode	Count down	FFFFH	Value if stop	Undefined	–
One-count mode	Count down	FFFFH	Value if stop	Undefined	FFFFH
Capture & one-count mode	Count up	0000H	Value if stop	Undefined	Capture value of 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 7)

### 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).

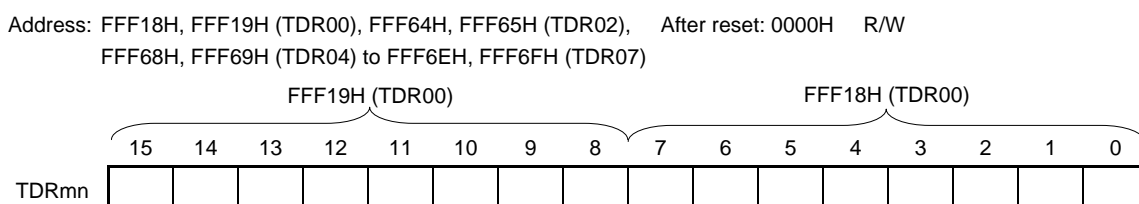
The value of the TDRmn register can be changed at any time.

This register can be read or written in 16-bit units.

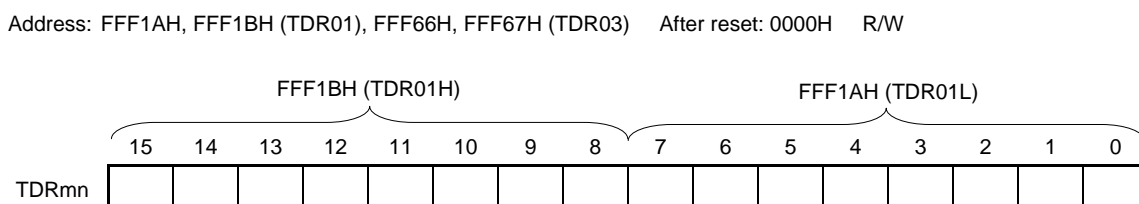
In addition, for the TDRm1 and TDRm3 registers, while in the 8-bit timer mode (when the SPLIT bits of timer mode registers 01 and 03 (TMRm1, TMRm3) are 1), it is possible to read and write data in 8-bit units, with the higher 8 bits used as TDRm1H and TDRm3H, and the lower 8 bits used as TDRm1L and TDRm3L.

Reset signal generation clears this register to 0000H.

**Figure 6-8. Format of Timer Data Register mn (TDRmn) (n = 0, 2, 4 to 7)**



**Figure 6-9. Format of Timer Data Register mn (TDRmn) (n = 1, 3)**



**(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 7)

### 6.3 Registers Controlling Timer Array Unit

The 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 select register (TOS)
- Timer output enable register m (TOEm)
- Timer output register m (TOM)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Input switch control register (ISC)
- Noise filter enable register 1 (NFEN1)
- Port mode control register (PMCxx)
- Port mode register (PMxx)
- Port register (Pxx)

**Caution** Which registers and bits are included depends on the product. Be sure to set bits that are not mounted to their initial values.

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

### 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 6-10. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

TAU0EN	Control of timer array unit input clock
0	Stops supply of input clock. <ul style="list-style-type: none"> <li>• SFR used by the timer array unit cannot be written.</li> <li>• The timer array unit is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the timer array unit can be read/written.</li> </ul>

**Cautions 1.** When setting the timer array unit, be sure to set the following registers first while the TAU0EN bit is set to 1. If TAU0EN = 0, the values of the registers which control the timer array unit are cleared to their initial values and writing to them is ignored (except for timer input select register 0 (TIS0), input switch control register (ISC), noise filter enable register 1, 2 (NFEN1, NFEN2), port mode control register 4 (PMC4), port mode register 0, 1, 3, 4, 5, 12 (PM0, PM1, PM3, PM4, PM5, PM12), port register 0, 1, 3, 4, 5, 12 (P0, P1, P3, P4, P5, P12)).

- 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)

2. Be sure to clear bits 1 and 6 to “0”.

### 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 7):

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 7):

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 6-11. Format of Timer Clock Select Register m (TPSm) (1/2)

Address: F01B6H, F01B7H 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) <sup>Note</sup> (k = 0, 1)	Selection of operation clock (CKmk) <sup>Note</sup> (k = 0, 1)				
					f <sub>CLK</sub> = 2 MHz	f <sub>CLK</sub> = 5 MHz	f <sub>CLK</sub> = 10 MHz	f <sub>CLK</sub> = 20 MHz	f <sub>CLK</sub> = 24 MHz
0	0	0	0	f <sub>CLK</sub>	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	0	0	1	f <sub>CLK</sub> /2	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	0	1	0	f <sub>CLK</sub> /2 <sup>2</sup>	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	125 kHz	313 kHz	625 kHz	1.25 MHz	1.50 MHz
0	1	0	1	f <sub>CLK</sub> /2 <sup>5</sup>	62.5 kHz	156 kHz	313 kHz	625 kHz	750 kHz
0	1	1	0	f <sub>CLK</sub> /2 <sup>6</sup>	31.3 kHz	78.1 kHz	156 kHz	313 kHz	375 kHz
0	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	188 kHz
1	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
1	0	0	1	f <sub>CLK</sub> /2 <sup>9</sup>	3.91 kHz	9.76 kHz	19.5 kHz	39.1 kHz	46.9 kHz
1	0	1	0	f <sub>CLK</sub> /2 <sup>10</sup>	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz	23.4 kHz
1	0	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	976 Hz	2.44 kHz	4.88 kHz	9.76 kHz	11.7 kHz
1	1	0	0	f <sub>CLK</sub> /2 <sup>12</sup>	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	0	1	f <sub>CLK</sub> /2 <sup>13</sup>	244 Hz	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	1	1	0	f <sub>CLK</sub> /2 <sup>14</sup>	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz
1	1	1	1	f <sub>CLK</sub> /2 <sup>15</sup>	61.0 Hz	153 Hz	305 Hz	610 Hz	732 Hz

**Note** When changing the clock selected for f<sub>CLK</sub> (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, and 10 to “0”.
  2. If f<sub>CLK</sub> (undivided) is selected as the operation clock (CKmk) and TDRnm is set to 0000H (n = 0, m = 0 to 7), interrupt requests output from timer array units cannot be used.

- Remarks**
1. f<sub>CLK</sub>: CPU/peripheral hardware clock frequency
  2. The above f<sub>CLK</sub>/2<sup>r</sup> is not a signal which is simply divided f<sub>CLK</sub> by 2<sup>r</sup>, but a signal which becomes high level for one period of f<sub>CLK</sub> from its rising edge (r = 1 to 15). For details, see 6.5.1 Count clock (f<sub>TCLK</sub>).

Figure 6-11. Format of Timer Clock Select Register m (TPSm) (2/2)

Address: F01B6H, F01B7H 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) <sup>Note</sup>					
		$f_{CLK} = 2 \text{ MHz}$	$f_{CLK} = 5 \text{ MHz}$	$f_{CLK} = 10 \text{ MHz}$	$f_{CLK} = 20 \text{ MHz}$	$f_{CLK} = 24 \text{ MHz}$	
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	313 kHz	625 MHz	1.25 MHz	1.5 MHz
1	1	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	375 kHz

PRS m31	PRS m30	Selection of operation clock (CKm3) <sup>Note</sup>					
		$f_{CLK} = 2 \text{ MHz}$	$f_{CLK} = 5 \text{ MHz}$	$f_{CLK} = 10 \text{ MHz}$	$f_{CLK} = 20 \text{ MHz}$	$f_{CLK} = 24 \text{ MHz}$	
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	5.86 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 ( $f_{MCK}$ ) specified by using the CKSmn0, and CKSmn1 bits or the valid edge of the signal input from the TImn pin is selected as the count clock ( $f_{TCLK}$ ).

**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 Clocks CKSm2 and CKSm3

Clock		Interval time <sup>Note</sup> ( $f_{CLK} = 20 \text{ MHz}$ )			
		16 $\mu\text{s}$	160 $\mu\text{s}$	1.6 ms	16 ms
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 the waveform of  $f_{CLK}/2^j$  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  $TE_{mn} = 1$ ). However, bits 7 and 6 ( $CIS_{mn1}$ ,  $CIS_{mn0}$ ) can be rewritten even while the register is operating with some functions (when  $TE_{mn} = 1$ ). (For details, see **6.8 Independent Channel Operation Function of Timer Array Unit** and **6.9 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, TMRm4, TMRm6: MASTERmn bit (n = 2, 4, 6)**

**TMRm1, TMRm3: SPLITmn bit (n = 1, 3)**

**TMRm0, TMRm5, TMRm7: Fixed to 0**



**Figure 6-12. Format of Timer Mode Register mn (TMRmn) (1/4)**

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) 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, 4, 6)	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, 5, 7)	CKS mn1	CKS mn0	0	CCS mn	0 <sup>Note</sup>	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 according to 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 read-only 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 = 00FFH) 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 7)

**Figure 6-12. Format of Timer Mode Register mn (TMRmn) (2/4)**

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) 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, 4, 6)	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, 5, 7)	CKS mn1	CKS mn0	0	CCS mn	0 <sup>Note</sup>	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

(Bit 11 of TMRmn (n = 2, 4, 6))

MAS TER mn	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, 4, 6 can be set as a master channel (MASTERmn = 1). Be sure to use channel 0, 5, 7 are 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))

SPLI Tmn	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.														

STS mn2	STS mn1	STS mn0	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 read-only and fixed to 0. Writing to this bit is ignored.

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

**Figure 6-12. Format of Timer Mode Register mn (TMRmn) (3/4)**

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) 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, 4, 6)	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, 5, 7)	CKS mn1	CKS mn0	0	CCS mn	0 <sup>Note</sup>	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	Operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer/Square wave output/PWM output (master)	Counting down
0	1	0	Capture mode	Input pulse interval measurement	Counting up
0	1	1	Event counter mode	External event counter	Counting down
1	0	0	One-count mode	Delay counter/One-shot pulse output/PWM output (slave)	Counting down
1	1	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 read-only and fixed to 0. Writing to this bit is ignored.

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

**Figure 6-12. Format of Timer Mode Register mn (TMRmn) (4/4)**

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) 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, 4, 6)	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, 5, 7)	CKS mn1	CKS mn0	0	CCS mn	0 <sup>Note 1</sup>	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 the previous page))	MD mn0	Setting of starting counting and interrupt
<ul style="list-style-type: none"> <li>Interval timer mode (0, 0, 0)</li> <li>Capture mode (0, 1, 0)</li> </ul>	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
<ul style="list-style-type: none"> <li>Event counter mode (0, 1, 1)</li> </ul>	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
<ul style="list-style-type: none"> <li>One-count mode<sup>Note 2</sup> (1, 0, 0)</li> </ul>	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated.
	1	Start trigger is valid during counting operation <sup>Note 3</sup> . At that time, interrupt is not generated.
<ul style="list-style-type: none"> <li>Capture &amp; one-count mode (1, 1, 0)</li> </ul>	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 read-only and fixed to 0. Writing to this bit is ignored.
- 2.** In one-count mode, interrupt output (INTTMmn) when starting a count operation and T0mn output are not controlled.
- 3.** If the start trigger (TSMn = 1) is issued during operation, the counter is initialized, an interrupt is generated, and recounting starts. (No interrupt request occurs.)

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

### 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 6-13. Format of Timer Status Register mn (TSRmn)**

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07) 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 7)

**Table 6-5. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode**

Timer Operation Mode	OVF Bit	Set/Clear Conditions
• Capture mode	Cleared	When no overflow has occurred upon capturing
• Capture & one-count mode	Set	When an overflow has occurred upon capturing
• Interval timer mode	Cleared	- (Use prohibited)
• Event counter mode	Set	
• One-count mode		

**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 (TE<sub>m</sub>)

The TE<sub>m</sub> register is used to enable or stop the timer operation of each channel.

Each bit of the TE<sub>m</sub> register corresponds to each bit of the timer channel start register m (T<sub>Sm</sub>) and the timer channel stop register m (T<sub>Tm</sub>). When a bit of the T<sub>Sm</sub> register is set to 1, the corresponding bit of this register is set to 1. When a bit of the T<sub>Tm</sub> register is set to 1, the corresponding bit of this register is cleared to 0.

The TE<sub>m</sub> register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TE<sub>m</sub> register can be set with a 1-bit or 8-bit memory manipulation instruction with TE<sub>mL</sub>.

Reset signal generation clears this register to 0000H.

**Figure 6-14. Format of Timer Channel Enable Status Register m (TE<sub>m</sub>)**

Address: F01B0H, F01B1H    After reset: 0000H    R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE <sub>m</sub>	0	0	0	0	TEH <sub>m3</sub>	0	TEH <sub>m1</sub>	0	TE <sub>m7</sub>	TE <sub>m6</sub>	TE <sub>m5</sub>	TE <sub>m4</sub>	TE <sub>m3</sub>	TE <sub>m2</sub>	TE <sub>m1</sub>	TE <sub>m0</sub>
					3		1		7	6	5	4	3	2	1	0

TEH <sub>m3</sub>	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.

TEH <sub>m1</sub>	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.

TE <sub>m</sub> <sub>n</sub>	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 TE <sub>m1</sub> and TE <sub>m3</sub> 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 7)

### 6.3.6 Timer channel start register m (T<sub>Sm</sub>)

The T<sub>Sm</sub> register is a trigger register that is used to initialize timer count register m<sub>n</sub> (TCR<sub>m<sub>n</sub></sub>) 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 (TE<sub>m</sub>) is set to 1. The T<sub>Sm<sub>n</sub></sub>, TSH<sub>m1</sub>, TSH<sub>m3</sub> bits are immediately cleared when operation is enabled (TE<sub>m<sub>n</sub></sub>, TEH<sub>m1</sub>, TEH<sub>m3</sub> = 1), because they are trigger bits.

The T<sub>Sm</sub> register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the T<sub>Sm</sub> register can be set with a 1-bit or 8-bit memory manipulation instruction with T<sub>SmL</sub>.

Reset signal generation clears this register to 0000H.

**Figure 6-15. Format of Timer Channel Start Register m (T<sub>Sm</sub>)**

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
T <sub>Sm</sub>	0	0	0	0	TSH <sub>m3</sub> 3	0	TSH <sub>m1</sub> 1	0	T <sub>Sm</sub> 7	T <sub>Sm</sub> 6	T <sub>Sm</sub> 5	T <sub>Sm</sub> 4	T <sub>Sm</sub> 3	T <sub>Sm</sub> 2	T <sub>Sm</sub> 1	T <sub>Sm</sub> 0

TSH <sub>m3</sub>	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 TEH <sub>m3</sub> bit is set to 1 and the count operation is enabled. The TCR <sub>m3</sub> register count operation start in the interval timer mode in the count operation enabled state (see <b>Table 6-6</b> in <b>6.5.2 Start timing of counter</b> ).

TSH <sub>m1</sub>	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 TEH <sub>m1</sub> bit is set to 1 and the count operation is enabled. The TCR <sub>m1</sub> register count operation start in the interval timer mode in the count operation enabled state (see <b>Table 6-6</b> in <b>6.5.2 Start timing of counter</b> ).

T <sub>Sm<sub>n</sub></sub>	Operation enable (start) trigger of channel n
0	No trigger operation
1	The TE <sub>m<sub>n</sub></sub> bit is set to 1 and the count operation is enabled. The TCR <sub>m<sub>n</sub></sub> register count operation start in the count operation enabled state varies depending on each operation mode (see <b>Table 6-6</b> in <b>6.5.2 Start timing of counter</b> ). This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for T <sub>Sm1</sub> and T <sub>Sm3</sub> when channel 1 or 3 is in the 8-bit timer mode.

- Cautions**
- Be sure to clear bits 15 to 12, 11, and 8 to “0”
  - When switching from a function that does not use T<sub>Im<sub>n</sub></sub> pin input to one that does, the following wait period is required from when timer mode register m<sub>n</sub> (TMR<sub>m<sub>n</sub></sub>) is set until the T<sub>Sm<sub>n</sub></sub> (TSH<sub>m1</sub>, TSH<sub>m3</sub>) bit is set to 1.  
 When the T<sub>Im<sub>n</sub></sub> pin noise filter is enabled (TNFEN<sub>m</sub> = 1): Four cycles of the operation clock (f<sub>MCK</sub>)  
 When the T<sub>Im<sub>n</sub></sub> pin noise filter is disabled (TNFEN<sub>m</sub> = 0): Two cycles of the operation clock (f<sub>MCK</sub>)

- Remarks**
- When the T<sub>Sm</sub> register is read, 0 is always read.
  - m: Unit number (m = 0), n: Channel number (n = 0 to 7)

### 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 6-16. 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	TTHm 3	0	TTHm 1	0	TTm 7	TTm 6	TTm 5	TTm 4	TTm 3	TTm 2	TTm 1	TTm 0

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, and 8 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 7)



### 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 6-17. 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 <sub>IL</sub> )
1	0	1	Subsystem clock (f <sub>SUB</sub> )
Other than above			Setting prohibited

**Caution** At least  $1/f_{MCK} + 10$  ns is necessary as the high-level and low-level widths of the timer input to be selected. Thus, the TIS02 bit cannot be set to 1 when f<sub>SUB</sub> is selected as f<sub>CLK</sub> (CSS in CKC register = 1).

### 6.3.9 Timer output select register (TOS)

The TOS register is used to enable the remote control output function.  
 Remote control output are generated by using the PWM output signal generated by channels 2 and 3 (mask wave) to mask the PWM output signal generated by channels 4 and 5 (carrier wave).  
 Rewriting the TOS register is only possible before counting starts (TE02, TE03, TE04, TE05 = 0).  
 The TOS register can be set by a 1-bit or 8-bit memory manipulation instruction.  
 Reset signal generation clears this register to 00H.

**Figure 6-18. Format of Timer Output Select Register (TOS)**

Address: F0079H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TOS	0	0	0	0	0	0	0	TOS0

TOS0	Remote control output setting
0	Disable (channels 2, 3, 4, and 5 are used for timer output)
1	Enable (remote control output to the REMOOUT pin)

**Caution** Channels 2, 3, 4, and 5 cannot be used for any other function when remote control output is enabled (TOS0 = 1).

**6.3.10 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 6-19. 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	TOE m7	TOE m6	TOE m5	TOE m4	TOE m3	TOE m2	TOE m1	TOE m0

TOE mn	Timer output enable/disable of channel n
0	Disable output of timer. Without reflecting on TOMn bit timer operation, to fixed the output. Writing to the TOMn bit is enabled and the level set in the TOMn bit is output from the TOMn pin.
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 8 to “0”.

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

**6.3.11 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 of 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 P52/TI00/TO00, P32/TI01/TO01, P54/TI02/TO02, P30/TI03/TO03, P14/TI04/TO04, P42/TI05/TO05, P56/TI06/TO06, or P15/TI07/TO07 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 6-20. 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	TOM 7	TOM 6	TOM 5	TOM 4	TOM 3	TOM 2	TOM 1	TOM 0

TOM n	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 8 to "0".

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

**6.3.12 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 6-21. 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	TOL m7	TOL m6	TOL m5	TOL m4	TOL m3	TOL m2	TOL m1	0

TOL mn	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 8, 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 7)

**6.3.13 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 6-22. 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	TOM m7	TOM m6	TOM m5	TOM m4	TOM m3	TOM m2	TOM m1	0

TOM mn	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 8, and 0 to “0”.

**Remark** m: Unit number (m = 0)  
 n: Channel number  
 n = 0 to 7 (n = 0, 2, 4, 6 for master channel)  
 p: Slave channel number  
 n < p ≤ 7

(For details of the relationship between the master channel and slave channel, see **6.4.1 Basic rules of simultaneous channel operation function.**)

### 6.3.14 Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to implement LIN-bus communication operation by using channel 7 in association with the serial array unit. When the ISC1 bit is set to 1, the input signal of the serial data input pin (RxD0) is selected as a timer input signal.

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

Reset signal generation clears this register to 00H.

**Figure 6-23. Format of Input Switch Control Register (ISC)**

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of the RxD0 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD0 pin as an external interrupt (wakeup signal detection).

- Cautions**
1. Be sure to clear bits 7 to 2 to "0".
  2. When using 16-bit timer KB20, set the ISC register to its initial value (00H).

**Remark** When the LIN-bus communication function is used, select the input signal of the RxD0 pin by setting ISC1 to 1.

### 6.3.15 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 enabled, after synchronization with the operating clock ( $f_{MCK}$ ) for the target channel, whether the signal keeps the same value for two clock cycles is detected.

When the noise filter is disabled, the input signal is only synchronized with the operating clock ( $f_{MCK}$ ) for the target channel<sup>Note</sup>.

The NFEN1 register 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)** and **6.5.2 Start timing of counter**.

Figure 6-24. 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	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00
TNFEN07	Enable/disable using noise filter of TI07 pin or RxD0 pin input signal <sup>Note</sup>							
0	Noise filter OFF							
1	Noise filter ON							
TNFEN06	Enable/disable using noise filter of TI06 pin input signal							
0	Noise filter OFF							
1	Noise filter ON							
TNFEN05	Enable/disable using noise filter of TI05 pin input signal							
0	Noise filter OFF							
1	Noise filter ON							
TNFEN04	Enable/disable using noise filter of TI04 pin input signal							
0	Noise filter OFF							
1	Noise filter ON							
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							

**Note** The applicable pin can be switched by setting the ISC1 bit of the ISC register.  
 ISC1 = 0: Whether or not to use the noise filter of the TI07 pin can be selected.  
 ISC1 = 1: Whether or not to use the noise filter of the RxD0 pin can be selected.



### 6.3.16 Registers controlling port functions of pins to be used for timer I/O

Using port pins for the timer array unit functions requires setting of the registers that control the port functions multiplexed on the target pins (port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx)). For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, and **4.3.6 Port mode control registers (PMCxx)**.

The port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx) to be set depend on the product. For details, see **4.5.3 Register setting examples for used port and alternate functions**.

When using the ports (such as P14/TI04/TO04 and P42/TI05/TO05) 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 P14/TI04/TO04 for timer output

Set the PM14 bit of port mode register 1 to 0.

Set the P14 bit of port register 1 to 0.

Set bit of LCD port function register (PFSEGx) to 0.

When using the ports (such as P14/TI04/TO04 and P42/TI05/TO05) 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 P14/TI04/TO04 for timer input

Set the PM14 bit of port mode register 1 to 1.

Set the P14 bit of port register 1 to 0 or 1.

Set bit of LCD port function register (PFSEGx) to 0.

## 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, 4, etc.) 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 or those that follow (channels 3, 4, 5, etc.) 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 4 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.

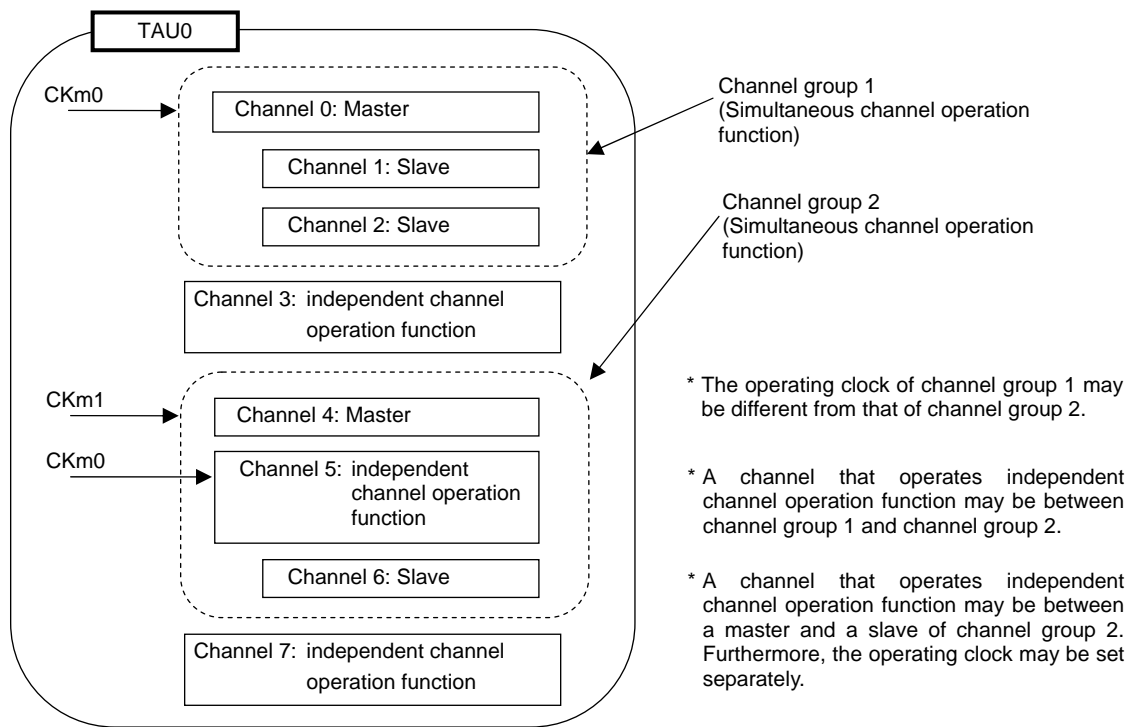
- (6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKSmn0, CKSmn1 bits (bit 15, 14 of timer mode register mn (TMRmn)) of the slave channel that operates in combination with the master channel must be the same value as 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 7)

Example



### 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 SPLIT 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
  - 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<sub>TCLK</sub>)

The count clock (f<sub>TCLK</sub>) of the timer array unit can be selected from the following according to the CCS<sub>mn</sub> bit of timer mode register mn (TMR<sub>mn</sub>).

- Operation clock (f<sub>MCK</sub>) specified by the CKS<sub>mn0</sub> and CKS<sub>mn1</sub> bits
- Valid edge of input signal input from the TI<sub>mn</sub> pin

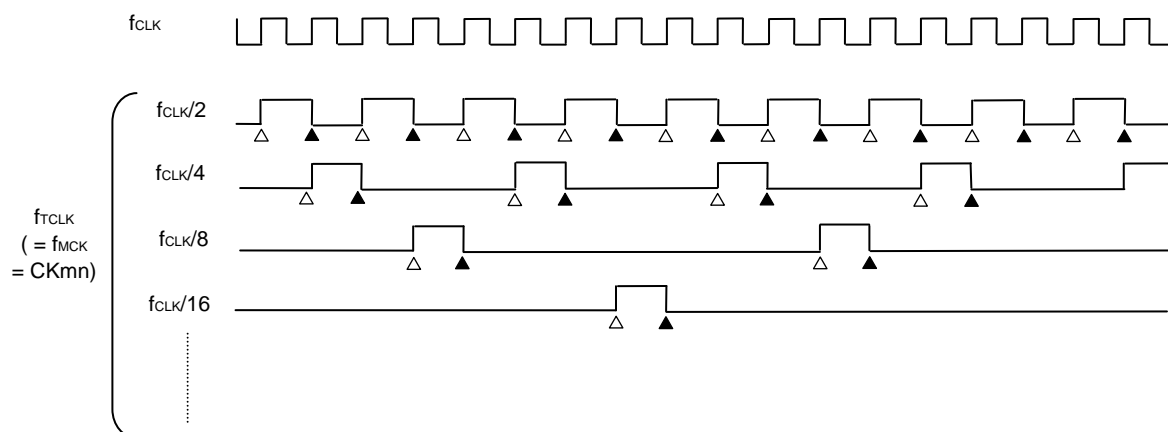
Because the timer array unit is designed to operate in synchronization with f<sub>CLK</sub>, the timings of the count clock (f<sub>TCLK</sub>) are shown below.

#### (1) When operation clock (f<sub>MCK</sub>) specified by the CKS<sub>mn0</sub> and CKS<sub>mn1</sub> bits is selected (CCS<sub>mn</sub> = 0)

The count clock (f<sub>TCLK</sub>) is between f<sub>CLK</sub> to f<sub>CLK</sub> / 2<sup>15</sup> by setting of timer clock select register m (TPS<sub>m</sub>). When a divided f<sub>CLK</sub> is selected, however, the clock selected in TPS<sub>mn</sub> register, but a signal which becomes high level for one period of f<sub>CLK</sub> from its rising edge. When a f<sub>CLK</sub> is selected, fixed to high level

Counting of timer count register mn (TCR<sub>mn</sub>) delayed by one period of f<sub>CLK</sub> from rising edge of the count clock, because of synchronization with f<sub>CLK</sub>. But, this is described as “counting at rising edge of the count clock”, as a matter of convenience.

Figure 6-25. Timing of f<sub>CLK</sub> and count clock (f<sub>TCLK</sub>) (When CCS<sub>mn</sub> = 0)



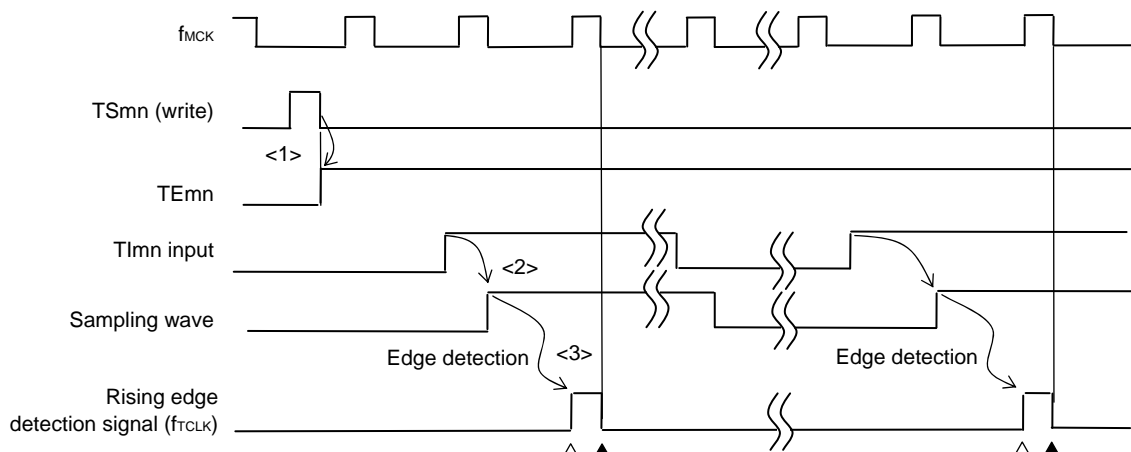
- Remarks 1.** Δ : Rising edge of the count clock  
 ▲ : Synchronization, increment/decrement of counter
- 2.** f<sub>CLK</sub>: CPU/peripheral hardware clock

**(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 6-26. 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 TImn pin input signal, which is used for input pulse interval measurement, input signal of high/low width measurement, the delay counter, and one-shot pulse output, is the same as that shown in above figure.

### 6.5.2 Start timing of counter

Operation of timer count register mn (TCRmn) is enabled by setting of TSmn bit of timer channel start register m (TSM). Operation from when counting is enabled to when timer count register mn (TCRmn) starts counting 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
<ul style="list-style-type: none"> <li>Interval timer mode</li> </ul>	<p>No operation is carried out from start trigger detection (TSmn=1) until count clock generation.</p> <p>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 <b>6.5.3 (1) Operation of interval timer mode</b>).</p>
<ul style="list-style-type: none"> <li>Event counter mode</li> </ul>	<p>Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register.</p> <p>If detect edge of TImn input. The subsequent count clock performs count down operation (see <b>6.5.3 (2) Operation of event counter mode</b>).</p>
<ul style="list-style-type: none"> <li>Capture mode</li> </ul>	<p>No operation is carried out from start trigger detection (TSmn = 1) until count clock generation.</p> <p>The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see <b>6.5.3 (3) Operation of capture mode (input pulse interval measurement)</b>).</p>
<ul style="list-style-type: none"> <li>One-count mode</li> </ul>	<p>The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0).</p> <p>No operation is carried out from start trigger detection until count clock generation.</p> <p>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 <b>6.5.3 (4) Operation of one-count mode</b>).</p>
<ul style="list-style-type: none"> <li>Capture &amp; one-count mode</li> </ul>	<p>The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0).</p> <p>No operation is carried out from start trigger detection until count clock generation.</p> <p>The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see <b>6.5.3 (5) Operation of capture &amp; one-count mode (high-level width measurement)</b>).</p>

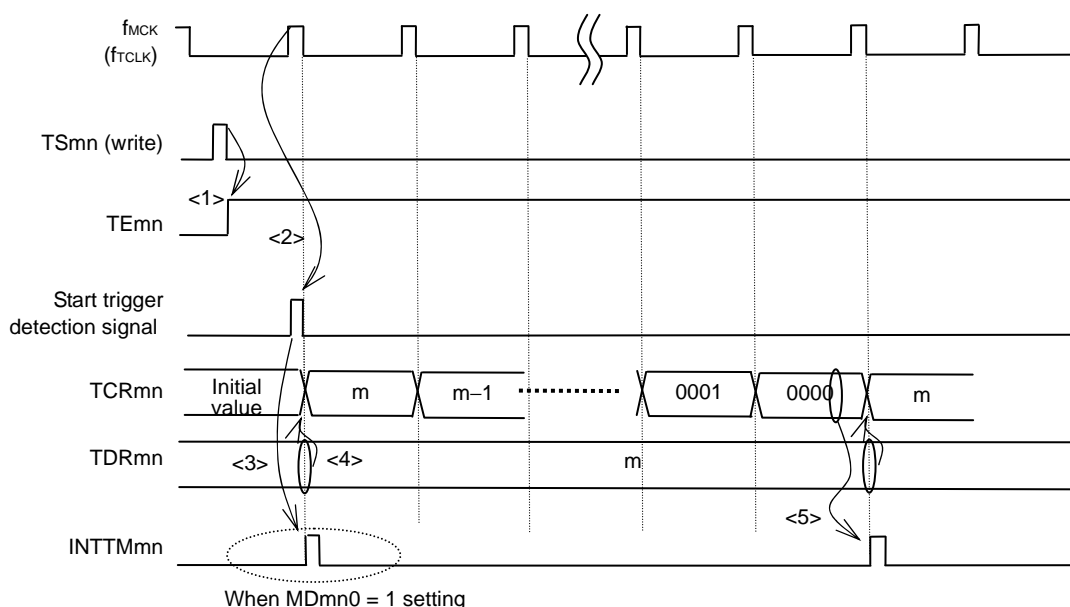
### 6.5.3 Operation of counter

Here, the counter operation in each mode is explained.

#### (1) Operation of interval timer mode

- <1> Operation is enabled ( $TE_{mn} = 1$ ) by writing 1 to the  $TS_{mn}$  bit. Timer count register  $mn$  ( $TCR_{mn}$ ) 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  $MD_{mn0}$  bit is set to 1,  $INTTM_{mn}$  is generated by the start trigger.
- <4> By the first count clock after the operation enable, the value of timer data register  $mn$  ( $TDR_{mn}$ ) is loaded to the  $TCR_{mn}$  register and counting starts in the interval timer mode.
- <5> When the  $TCR_{mn}$  register counts down and its count value is 0000H,  $INTTM_{mn}$  is generated and the value of timer data register  $mn$  ( $TDR_{mn}$ ) is loaded to the  $TCR_{mn}$  register and counting keeps on.

Figure 6-27. Operation Timing (In Interval Timer Mode)



**Caution** In the operation in the first count clock cycle after writing the  $TS_{mn}$  bit, an error at a maximum of one count clock cycle occurs since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated when counting is started by setting  $MD_{mn0} = 1$ .

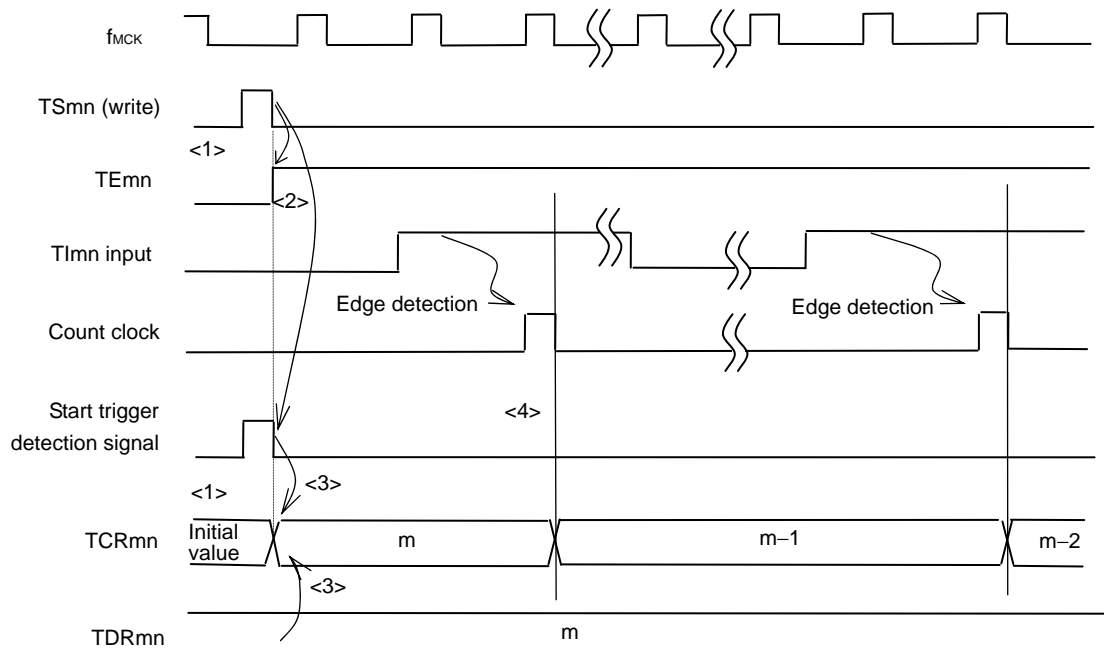
**Remark**  $f_{MCK}$ , the start trigger detection signal, and  $INTTM_{mn}$  become active for one clock cycle in synchronization with  $f_{CLK}$ .



**(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 6-28. Operation Timing (In Event Counter Mode)**

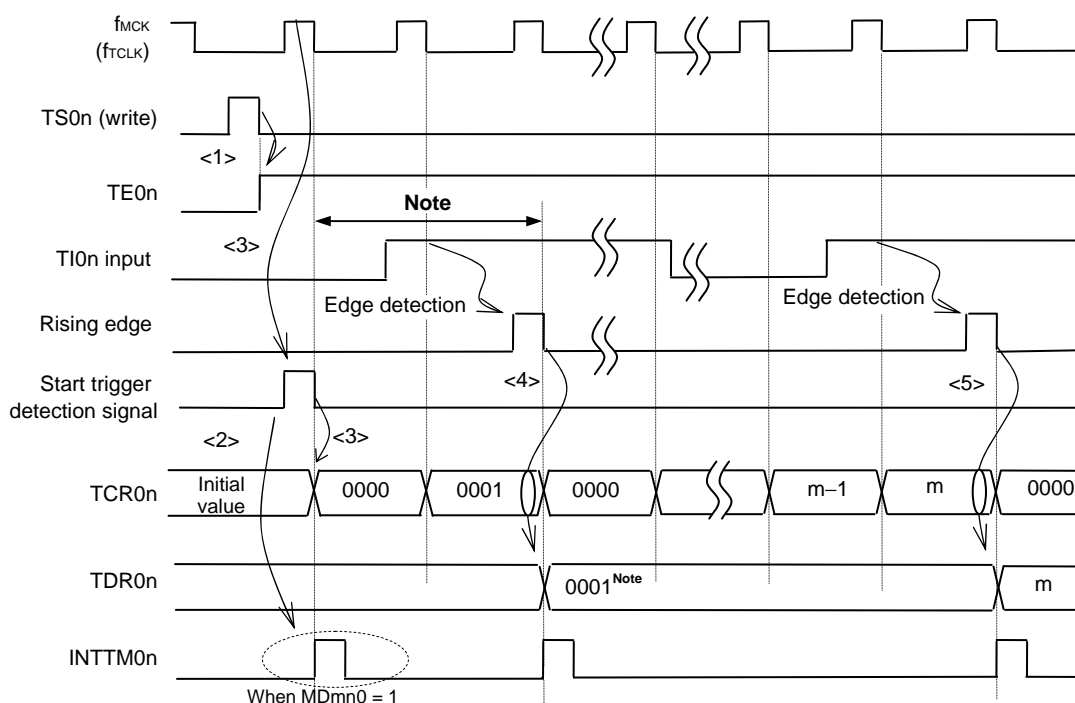


**Remark** The above figure indicates the timing when the noise filter is not used. When the noise filter is turned on, edge detection is delayed by 2 *f<sub>MCK</sub>* cycles (it sums up to 3 to 4 cycles) from TImn input. The error per one cycle occurs because the TImn input is not synchronous with the count clock (*f<sub>MCK</sub>*).

**(3) Operation of capture mode (input pulse interval measurement)**

- <1> Operation is enabled ( $TE_{mn} = 1$ ) by writing 1 to the  $TS_{mn}$  bit.
- <2> Timer count register  $mn$  ( $TCR_{mn}$ ) 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  $TCR_{mn}$  register and counting starts in the capture mode. (When the  $MD_{mn0}$  bit is set to 1,  $INTTM_{mn}$  is generated by the start trigger.)
- <4> On detection of the valid edge of the  $TI_{mn}$  input, the value of the  $TCR_{mn}$  register is captured to timer data register  $mn$  ( $TDR_{mn}$ ) and  $INTTM_{mn}$  is generated. However, this capture value is no meaning. The  $TCR_{mn}$  register keeps on counting from 0000H.
- <5> On next detection of the valid edge of the  $TI_{mn}$  input, the value of the  $TCR_{mn}$  register is captured to timer data register  $mn$  ( $TDR_{mn}$ ) and  $INTTM_{mn}$  is generated.

**Figure 6-29. Operation Timing (In Capture Mode: Input Pulse Interval Measurement)**



**Note** If a clock has been input to  $TI_{mn}$  (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.

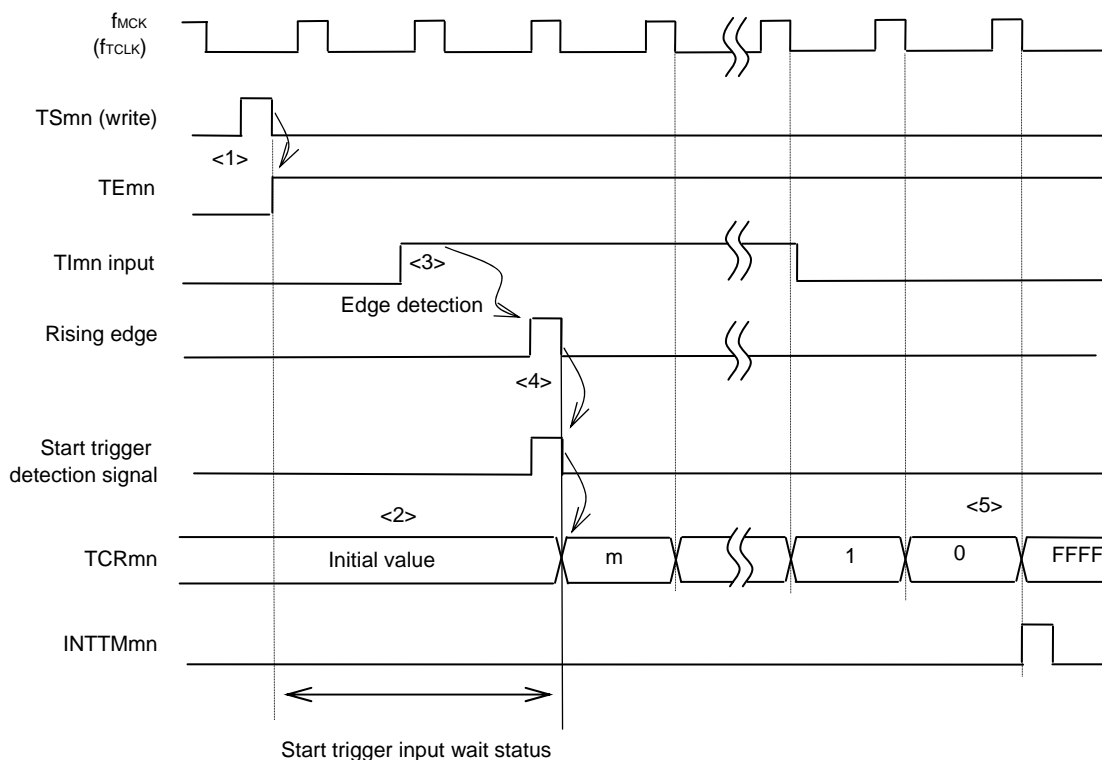
**Caution** In the operation in the first count clock cycle after writing the  $TS_{mn}$  bit, an error at a maximum of one count clock cycle occurs since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated when counting is started by setting  $MD_{mn0} = 1$ .

**Remark** The above figure indicates the timing when the noise filter is not used. When the noise filter is turned on, edge detection is delayed by 2  $f_{MCK}$  cycles (it sums up to 3 to 4 cycles) from  $TI_{mn}$  input. The error per one cycle occurs because the  $TI_{mn}$  input is not synchronous with the count clock ( $f_{MCK}$ ).

**(4) Operation of one-count mode**

- <1> Operation is enabled ( $TE_{mn} = 1$ ) by writing 1 to the  $TS_{mn}$  bit.
- <2> Timer count register  $mn$  ( $TCR_{mn}$ ) holds the initial value until start trigger generation.
- <3> Rising edge of the  $TI_{mn}$  input is detected.
- <4> On start trigger detection, the value of timer data register  $mn$  ( $TDR_{mn}$ ) is loaded to the  $TCR_{mn}$  register and count starts.
- <5> When the  $TCR_{mn}$  register counts down and its count value is 0000H,  $INTTM_{mn}$  is generated and the value of the  $TCR_{mn}$  register becomes FFFFH and counting stops.

**Figure 6-30. Operation Timing (In One-count Mode)**

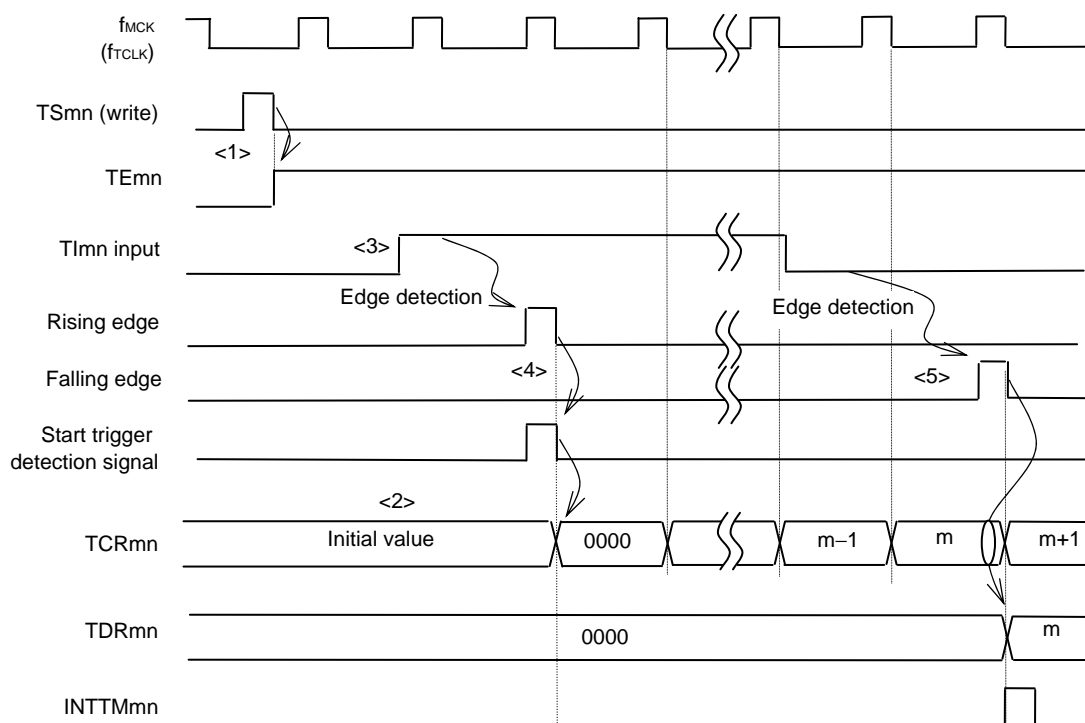


**Remark** The above figure indicates the timing when the noise filter is not used. When the noise filter is turned on, edge detection is delayed by 2  $f_{MCK}$  cycles (it sums up to 3 to 4 cycles) from  $TI_{mn}$  input. The error per one cycle occurs because the  $TI_{mn}$  input is not synchronous with the count clock ( $f_{MCK}$ ).

**(5) Operation of capture & one-count mode (high-level width measurement)**

- <1> Operation is enabled ( $TE_{mn} = 1$ ) by writing 1 to the  $TS_{mn}$  bit of timer channel start register  $m$  ( $TS_m$ ).
- <2> Timer count register  $mn$  ( $TCR_{mn}$ ) holds the initial value until start trigger generation.
- <3> Rising edge of the  $TI_{mn}$  input is detected.
- <4> On start trigger detection, the value of 0000H is loaded to the  $TCR_{mn}$  register and count starts.
- <5> On detection of the falling edge of the  $TI_{mn}$  input, the value of the  $TCR_{mn}$  register is captured to timer data register  $mn$  ( $TDR_{mn}$ ) and  $INTT_{mn}$  is generated.

**Figure 6-31. Operation Timing (In Capture & One-count Mode: High-level Width Measurement)**

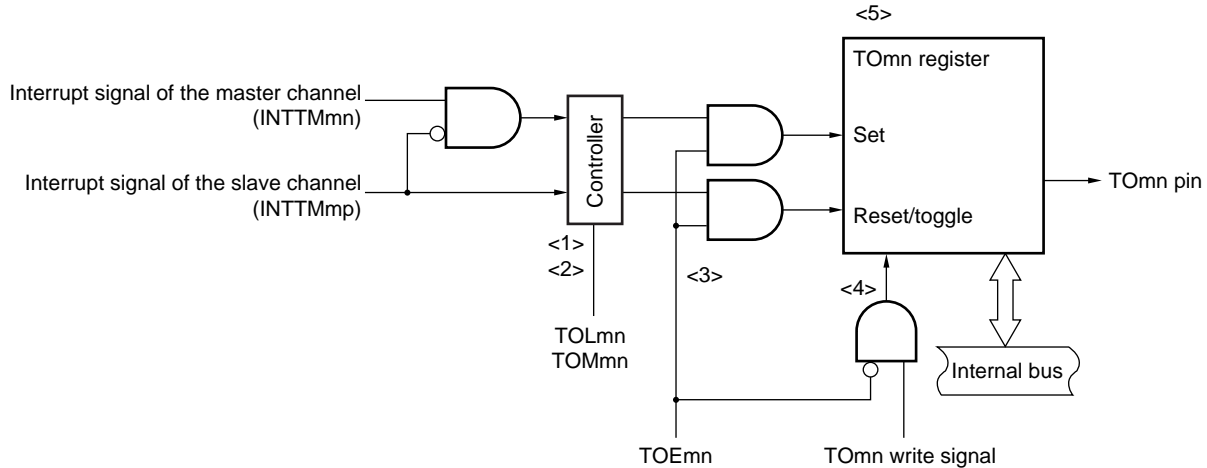


**Remark** The above figure indicates the timing when the noise filter is not used. When the noise filter is turned on, edge detection is delayed by 2  $f_{MCK}$  cycles (it sums up to 3 to 4 cycles) from  $TI_{mn}$  input. The error per one cycle occurs because the  $TI_{mn}$  input is not synchronous with the count clock ( $f_{MCK}$ ).

## 6.6 Channel Output (TOMn pin) Control

### 6.6.1 TOMn pin output circuit configuration

Figure 6-32. 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 INTTM0p (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 INTTM0p (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: Positive logic output (INTTMmn → set, INTTM0p → reset)
- When TOLmn = 1: Negative logic output (INTTMmn → reset, INTTM0p → set)

When INTTMmn and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTMmn (set signal) is masked.

- <3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTM0p (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 stopped (TOEmn = 0) and to write a value to the TOM register.
- <4> While timer output is disabled (TOEmn = 0), writing to the TOMn bit to the target channel (TOMn write signal) becomes valid. When timer output is disabled (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.

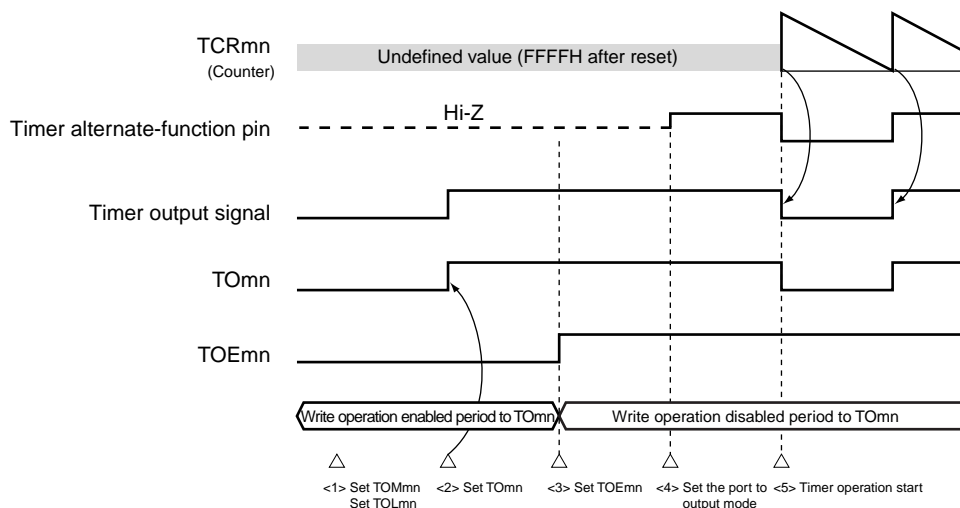
**Caution** Since outputs are N-ch open-drain outputs, an external pull-up resistor is required to use P60 and P61 as channel output.

**Remark** m: Unit number (m = 0)  
n: Channel number  
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)  
p: Slave channel number  
n < p ≤ 7

### 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 6-33. 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).

<5> The port I/O setting is set to output (see **6.3.16 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 7)

### 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 6.8 and 6.9.

When the values set to the TOEm, and TOMm 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 7)

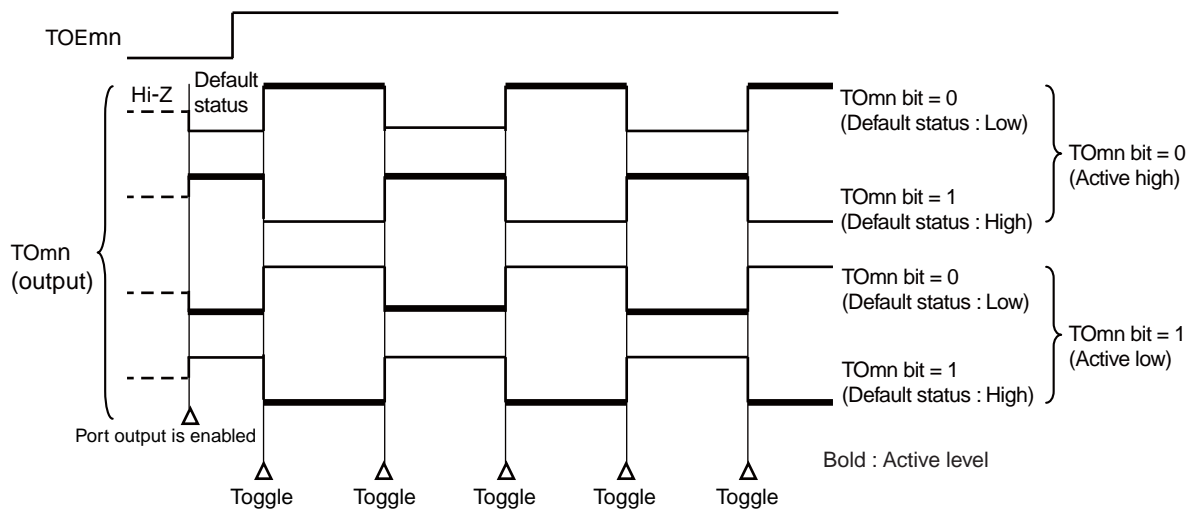
**(2) Default level of TOMn pin and output level after timer operation start**

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 6-34. TOMn Pin Output Status at Toggle Output (TOMmn = 0)**



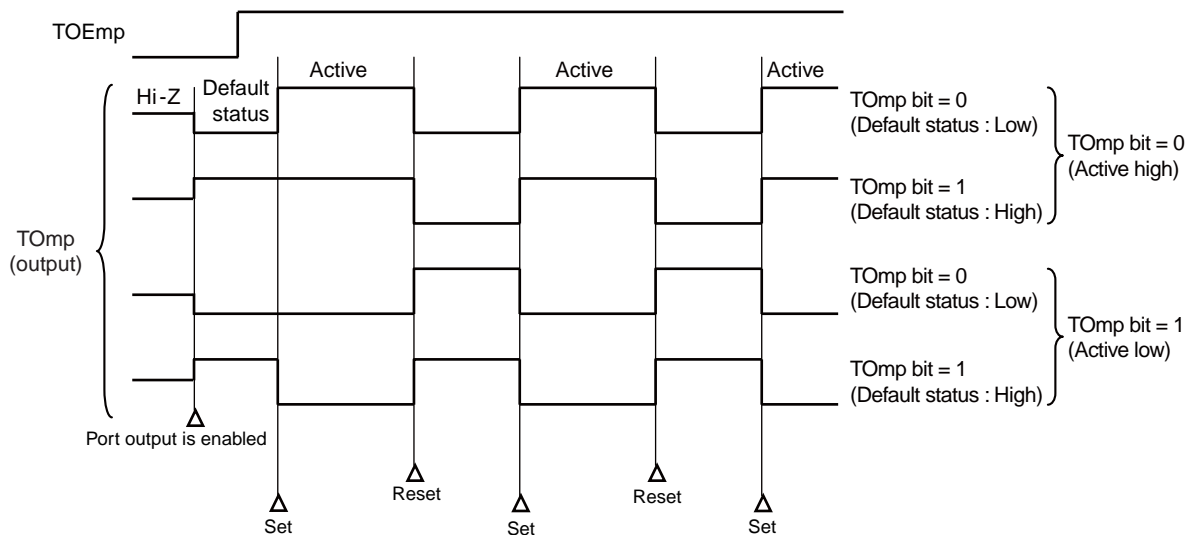
- Remarks 1.** Toggle: Reverse TOMn pin output status  
**2.** m: Unit number (m = 0), n: Channel number (n = 0 to 7)



**(b) When operation starts with slave channel output mode (TOMmp = 1) setting (PWM output)**

When slave channel output mode (TOMmp = 1), the active level is determined by timer output level register m (TOLm) setting.

**Figure 6-35. TOmp Pin Output Status at PWM Output (TOMmp = 1)**



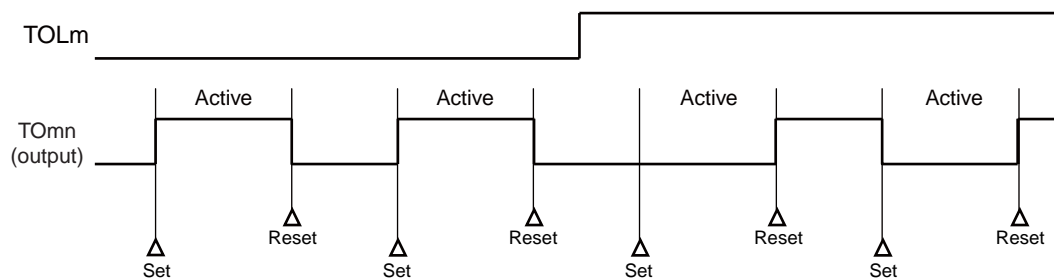
- Remarks 1.** Set: The output signal of the TOmp pin changes from inactive level to active level.  
 Reset: The output signal of the TOmp pin changes from active level to inactive level.
- 2.** m: Unit number (m = 0), p: Channel number (p = 1 to 7)

**(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 6-36. 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 7)

**(b) Set/reset timing**

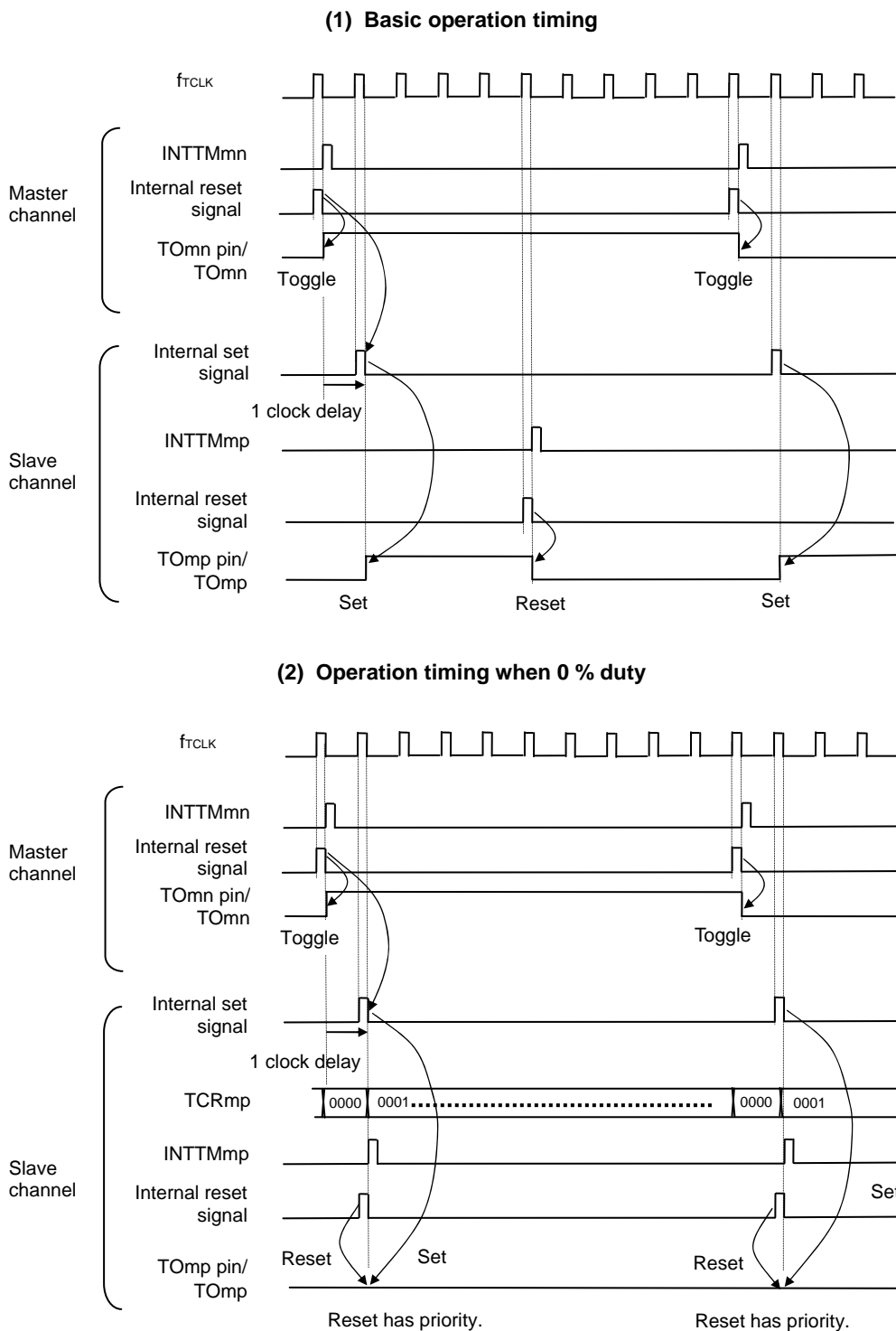
To realize 0%/100% output at PWM output, the TOMn pin/TOMn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-37 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 6-37. Set/Reset Timing Operating Statuses



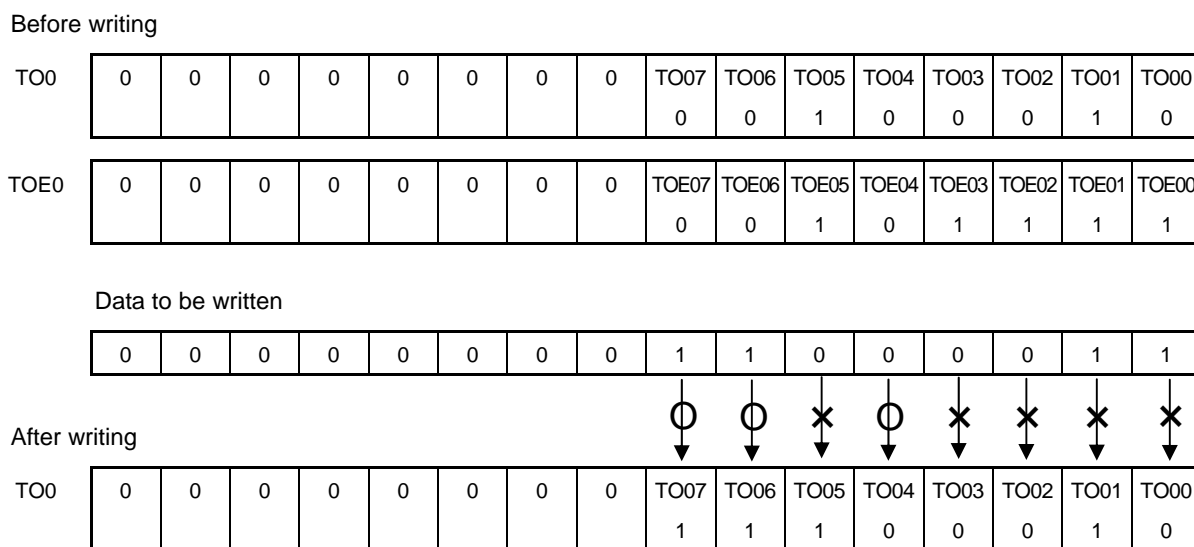
- Remarks 1.** Internal reset signal: TOmn pin reset/toggle signal  
 Internal set signal: TOmn pin set signal
- 2.** m: Unit number (m = 0)  
 n: Channel number  
 n = 0 to 7 (n = 0, 2, 4, 6 for master channel)  
 p: Slave channel number  
 n < p ≤ 7

### 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).

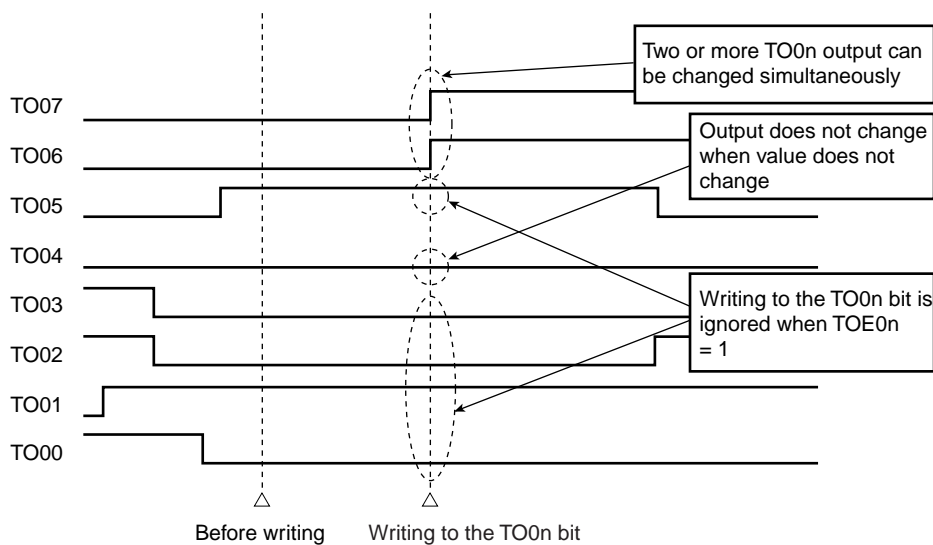
Figure 6-38 Example of TO0n Bit Collective Manipulation



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 6-39. TO0n Pin Statuses by Collective Manipulation of TO0n Bit



**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

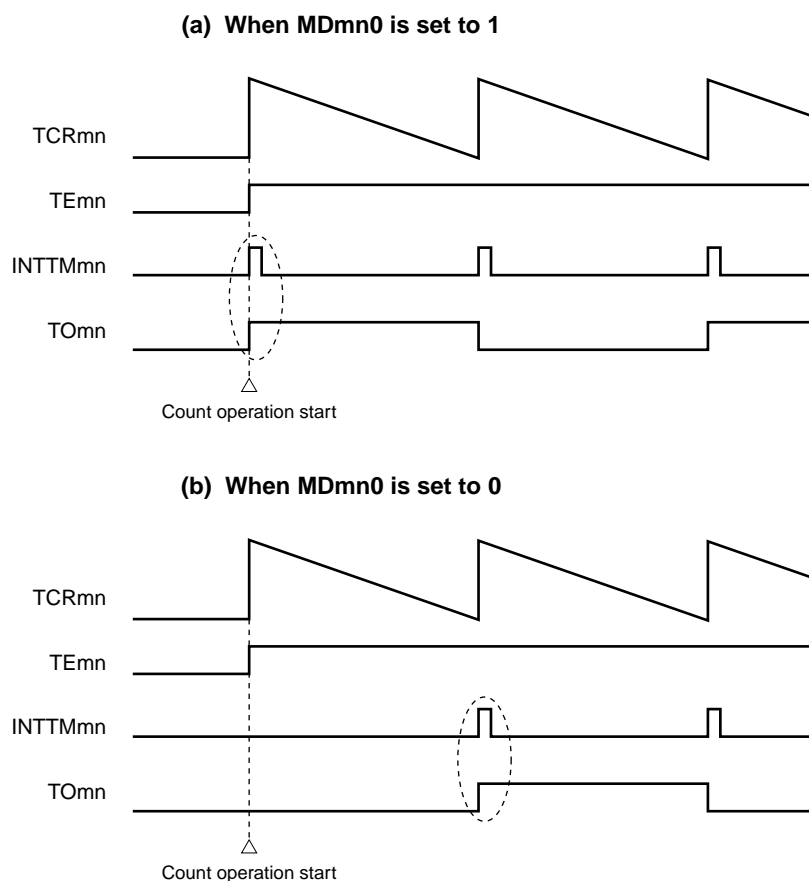
### 6.6.5 Timer interrupt and TOMn pin output at operation start

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 6-40 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

**Figure 6-40. Operation Examples of Timer Interrupt at Count Operation Start and TOMn Output**



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.

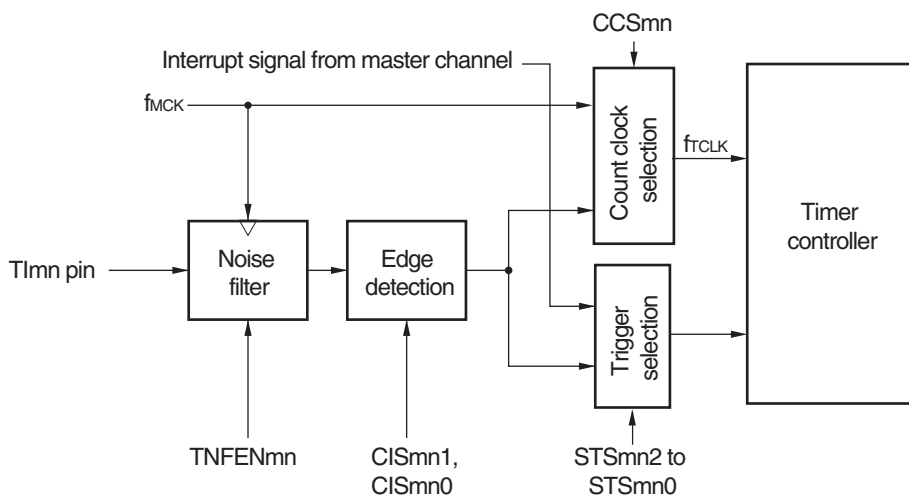
**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

### 6.7 Timer Input (Tlmn) Control

#### 6.7.1 Tlmn input circuit configuration

A signal is input from a timer input pin, goes through a noise filter and an edge detector, and is sent to a timer controller. Enable the noise filter for the pin in need of noise removal. The following shows the configuration of the input circuit.

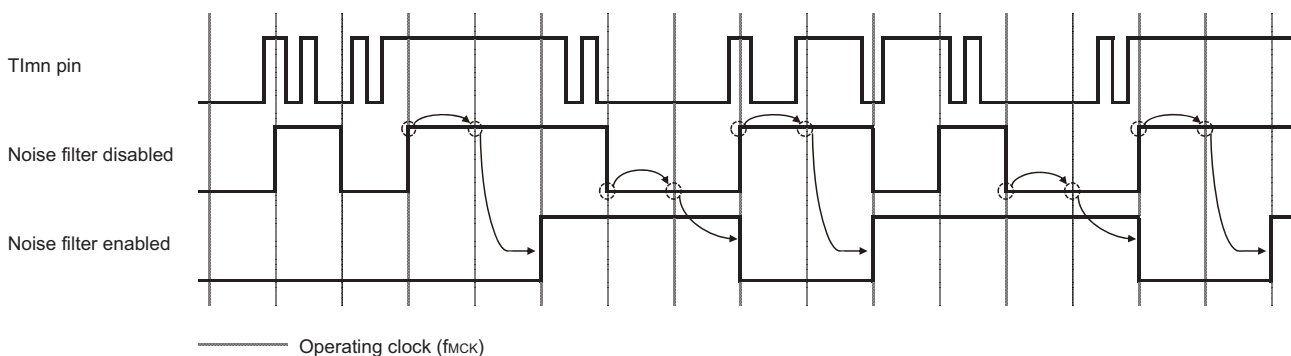
Figure 6-41. Input Circuit Configuration



#### 6.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock ( $f_{MCK}$ ) for channel  $n$ . When the noise filter is enabled, after synchronization with the operating clock ( $f_{MCK}$ ) for channel  $n$ , whether the signal keeps the same value for two clock cycles is detected. The following shows differences in waveforms output from the noise filter between when the noise filter is enabled and disabled.

Figure 6-42. Sampling Waveforms through Tlmn Input Pin with Noise Filter Enabled and Disabled



**Caution** The Tlmn pin input waveform is shown to explain the noise filter ON/OFF operation. For actual operation, refer to the high-level width/low-level width in 32.4 or 33.4 AC Characteristics.

### 6.7.3 Cautions on channel input operation

When a timer input pin is set as unused, the operating clock is not supplied to the noise filter. Therefore, after settings are made to use the timer input pin, the following wait time is necessary before a trigger is specified to enable operation of the channel corresponding to the timer input pin.

(1) Noise filter is disabled

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are 0 and then one of them is set to 1, wait for at least two cycles of the operating clock ( $f_{MCK}$ ), and then set the operation enable trigger bit in the timer channel start register (TSM).

(2) Noise filter is enabled

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are all 0 and then one of them is set to 1, wait for at least four cycles of the operating clock ( $f_{MCK}$ ), and then set the operation enable trigger bit in the timer channel start register (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.

<R> 
$$\text{Generation period of INTTMmn (timer interrupt)} = \text{Period of count clock} \times (\text{Set value of TDRmn} + 1)$$

#### (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.

- $$\text{Period of square wave output from TOMn} = \text{Period of count clock} \times (\text{Set value of TDRmn} + 1) \times 2$$

- $$\text{Frequency of square wave output from TOMn} = \text{Frequency of count clock} / \{(\text{Set value of TDRmn} + 1) \times 2\}$$

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.

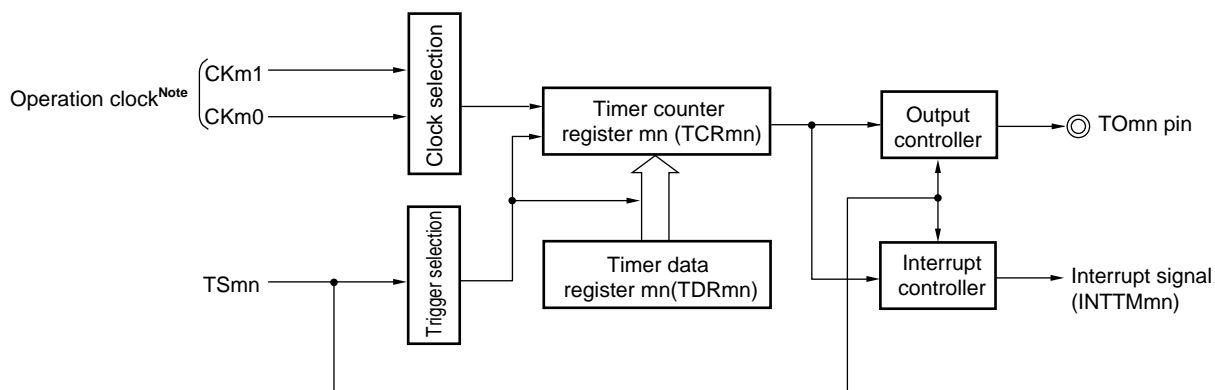
When TCRmn = 0000H, INTTMmn is output and TOMn is toggled at the next count clock. At the same time, the TCRmn register loads the value of the TDRmn register again. After that, the same operation is repeated.

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 7)

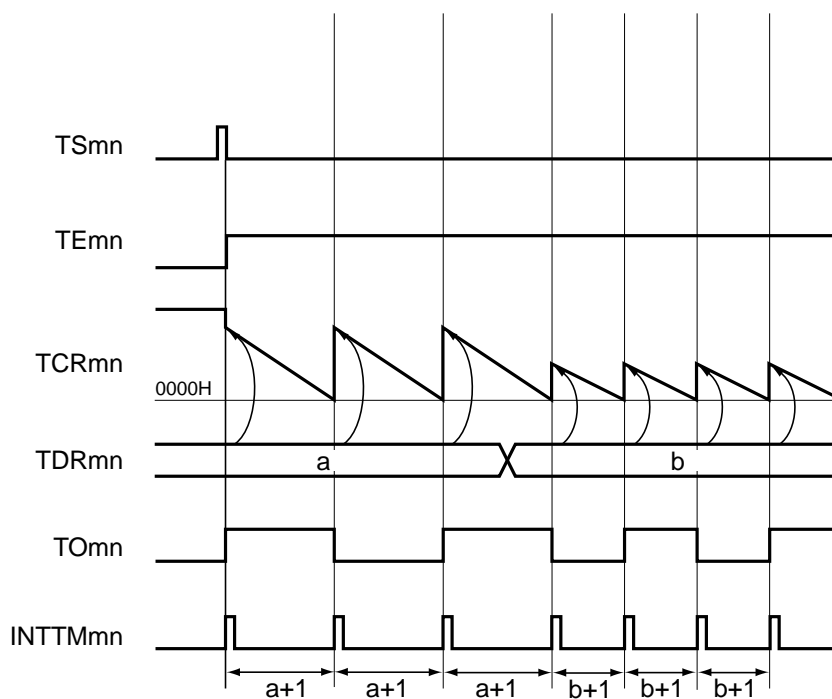


Figure 6-43. Block Diagram of Operation as Interval Timer/Square Wave Output



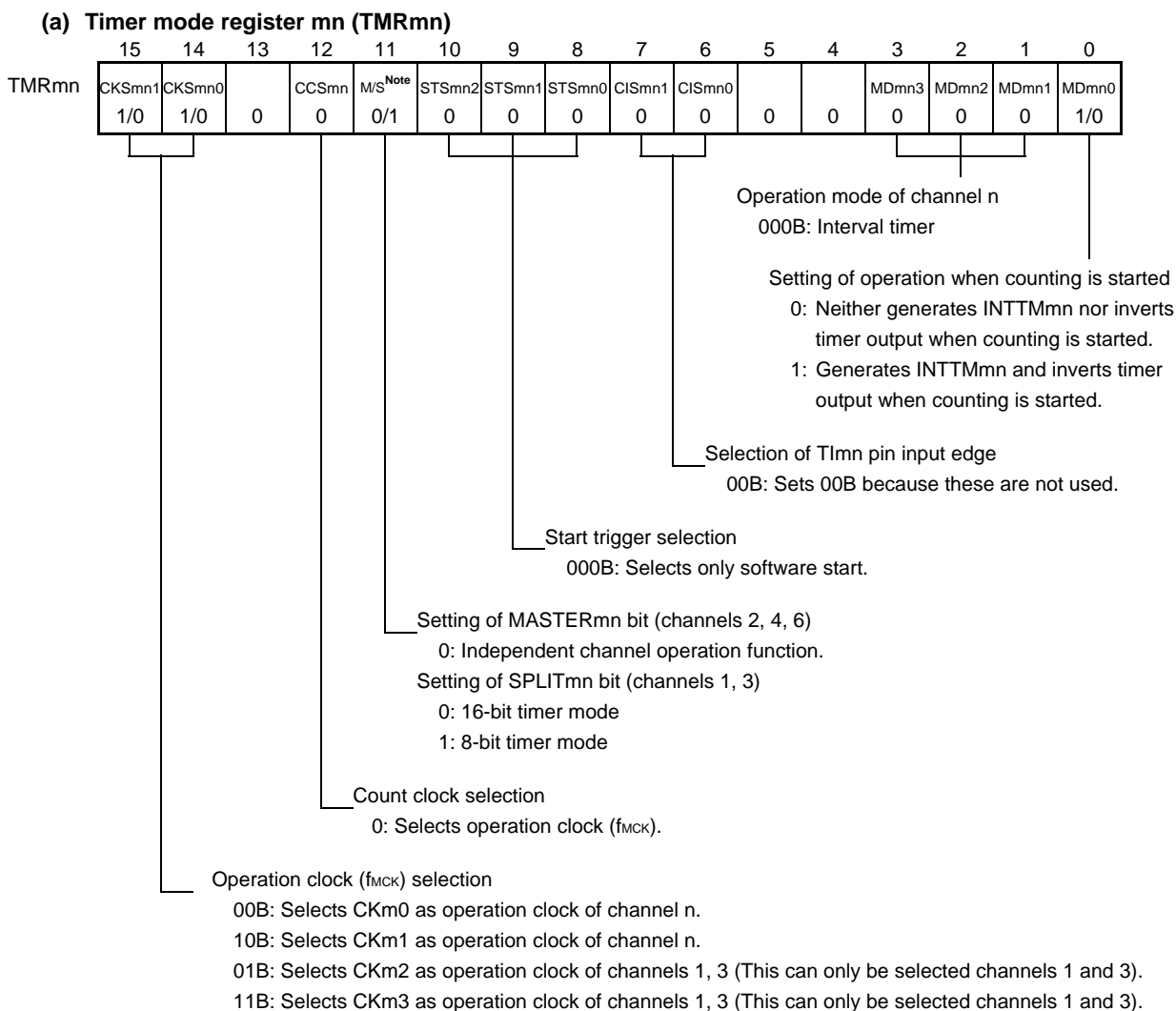
**Note** When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6-44. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)

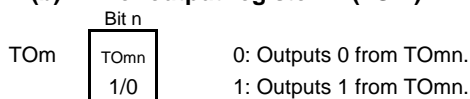


- Remarks**
1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)
  2. TSmn: Bit n of timer channel start register m (TSm)
  - TEmn: Bit n of timer channel enable status register m (TEm)
  - TCRmn: Timer count register mn (TCRmn)
  - TDRmn: Timer data register mn (TDRmn)
  - TOn: TOn pin output signal

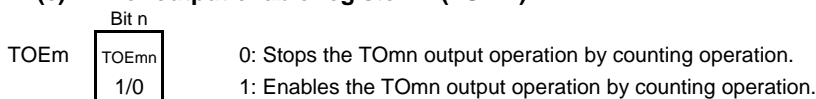
Figure 6-45. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)



(b) Timer output register m (TOM)



(c) Timer output enable register m (TOEm)



**Note** TMRm2, TMRm4, TMRm6: MASTERmn bit  
 TMRm1, TMRm3: SPLITmn bit  
 TMRm0, TMRm5, TMRm7: Fixed to 0

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

**Figure 6-45. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/2)****(d) Timer output level register m (TOLm)**

TOLm 

Bit n
TOLmn
0

 0: Cleared to 0 when TOMmn = 0 (master channel output mode)

**(e) Timer output mode register m (TOMm)**

TOMm 

Bit n
TOMmn
0

 0: Sets master channel output mode.

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-46. 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 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 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. Clears the port register and port mode register to 0.	TOMn does not change because channel stops operating. 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) at the count clock input. INTTMmn is generated and TOMn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
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 TOM and TOEm registers can be changed. Set values of the TMRmn register, TOMmn, and TOLmn bits cannot 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.

(Remark is listed on the next page.)

**Figure 6-46. 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. —————> When holding the TOMn pin output level is not necessary Setting not required. ----- The TAUmEN bit of the PER0 register is cleared to 0. —>	The TOMn pin output level is held by port function.  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 7)

**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.

$$\text{Specified number of counts} = \text{Set value of TDRmn} + 1$$

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, TSHm1, TSHm3) of timer channel start register m (TSM) to 1.

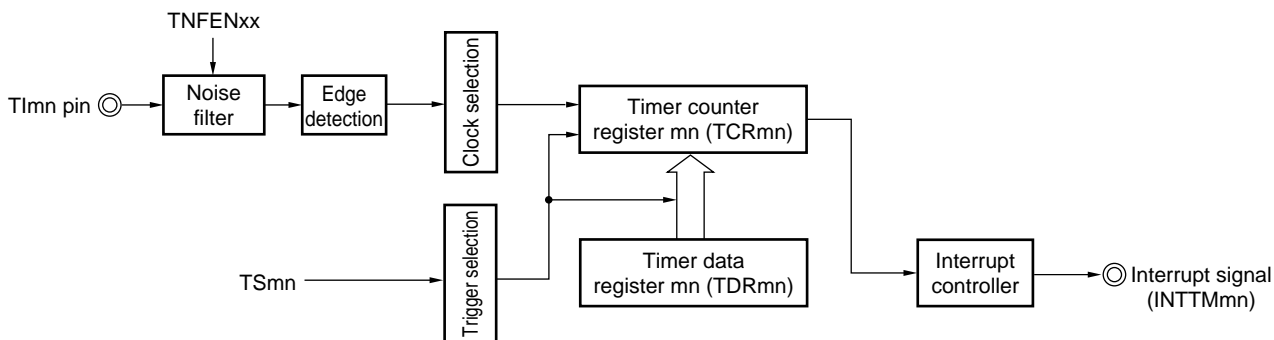
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.

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.

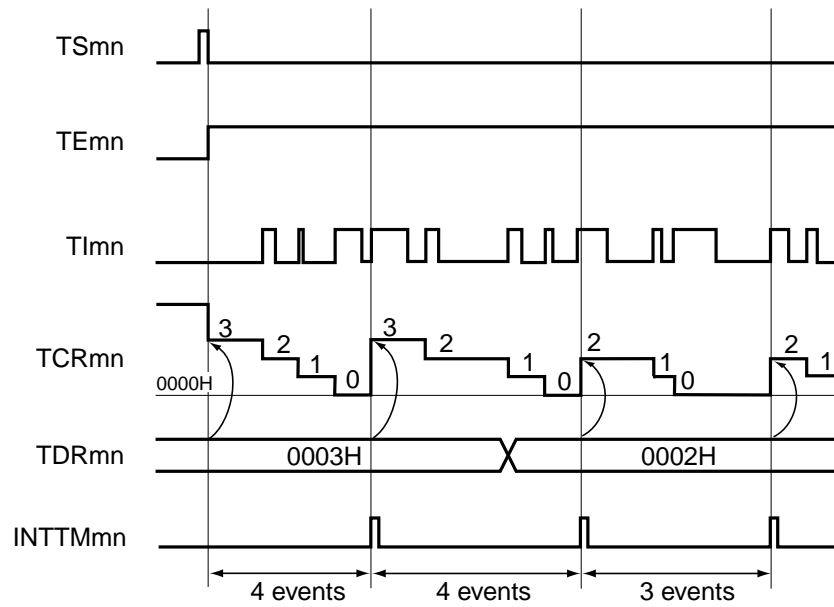
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

**Figure 6-47. Block Diagram of Operation as External Event Counter**



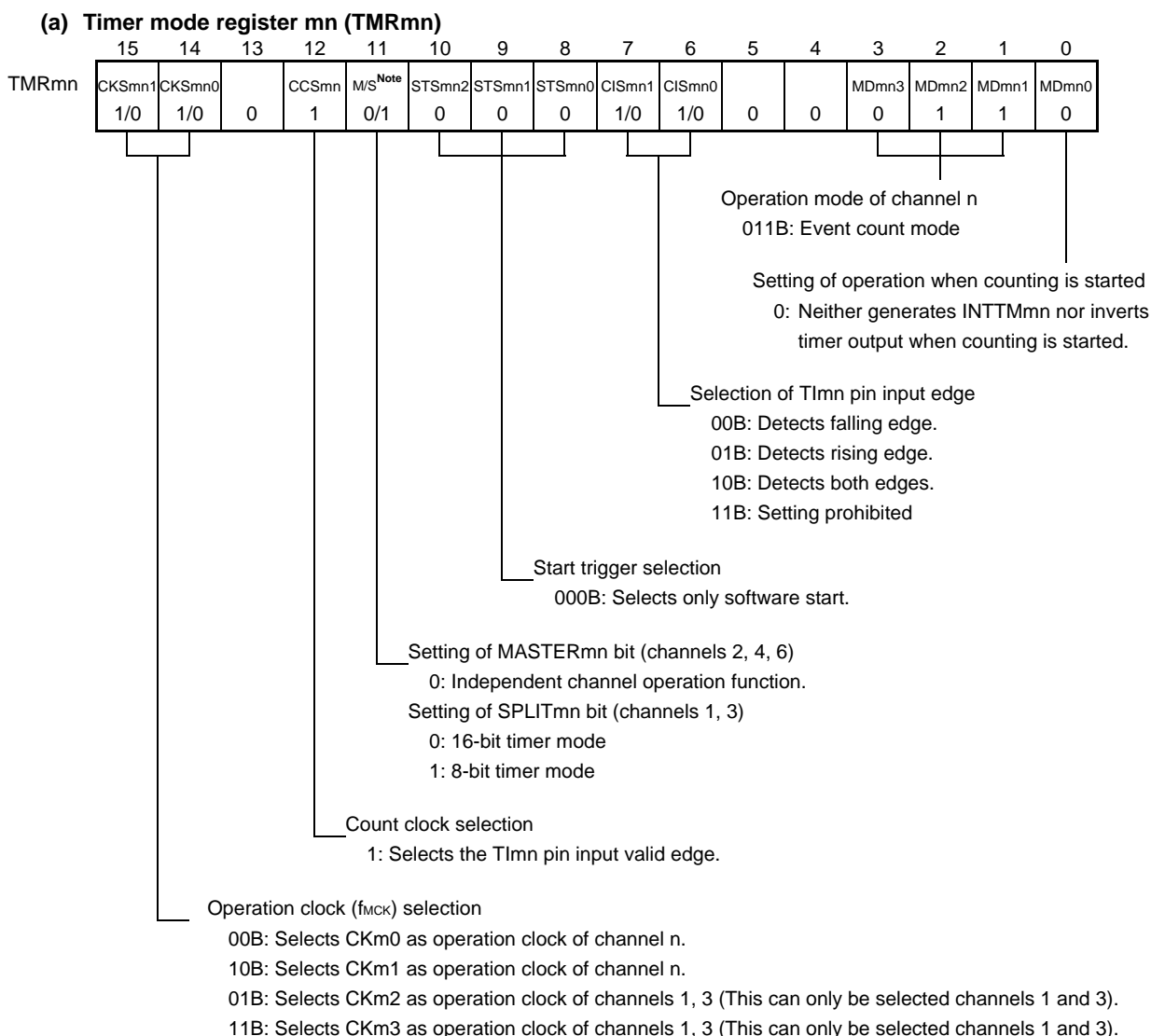
**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-48. Example of Basic Timing of Operation as External Event Counter

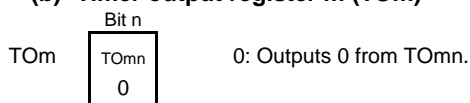


- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0 to 7)
  - TSmn: Bit n of timer channel start register m (TSM)
    - TE mn: Bit n of timer channel enable status register m (TEM)
    - TImn: TImn pin input signal
    - TCRmn: Timer count register mn (TCRmn)
    - TDRmn: Timer data register mn (TDRmn)

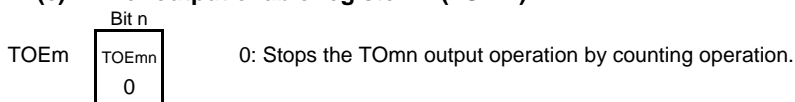
Figure 6-49. Example of Set Contents of Registers in External Event Counter Mode (1/2)



(b) Timer output register m (TOM)



(c) Timer output enable register m (TOEm)



**Note** TMRm2, TMRm4, TMRm6: MASTERmn bit  
 TMRm1, TMRm3: SPLITmn bit  
 TMRm0, TMRm5, TMRm7: Fixed to 0

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)



**Figure 6-49. Example of Set Contents of Registers in External Event Counter Mode (2/2)****(d) Timer output level register m (TOLm)**

TOLm 

Bit n
TOLmn
0

 0: Cleared to 0 when TOMmn = 0 (master channel output mode).

**(e) Timer output mode register m (TOMm)**

TOMm 

Bit n
TOMmn
0

 0: Sets master channel output mode.

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-50. 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 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). 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. The TSmn bit automatically returns to 0 because it is a trigger bit.	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. 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 7)

### 6.8.3 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.

$$\text{TImn input pulse interval} = \text{Period of count clock} \times ((10000\text{H} \times \text{TSRmn: OVF}) + (\text{Capture value of TDRmn} + 1))$$

**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.

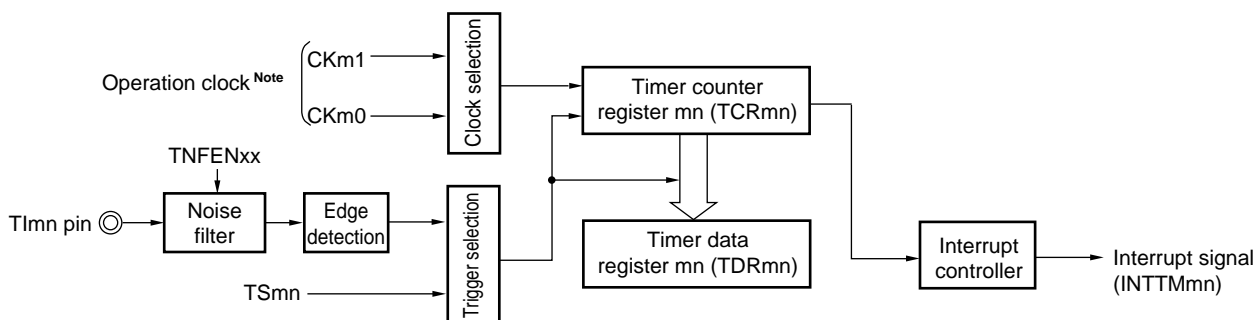
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

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.

When TEMn = 1, a software operation (TSmn = 1) can be used as a capture trigger, instead of using the TImn pin input.

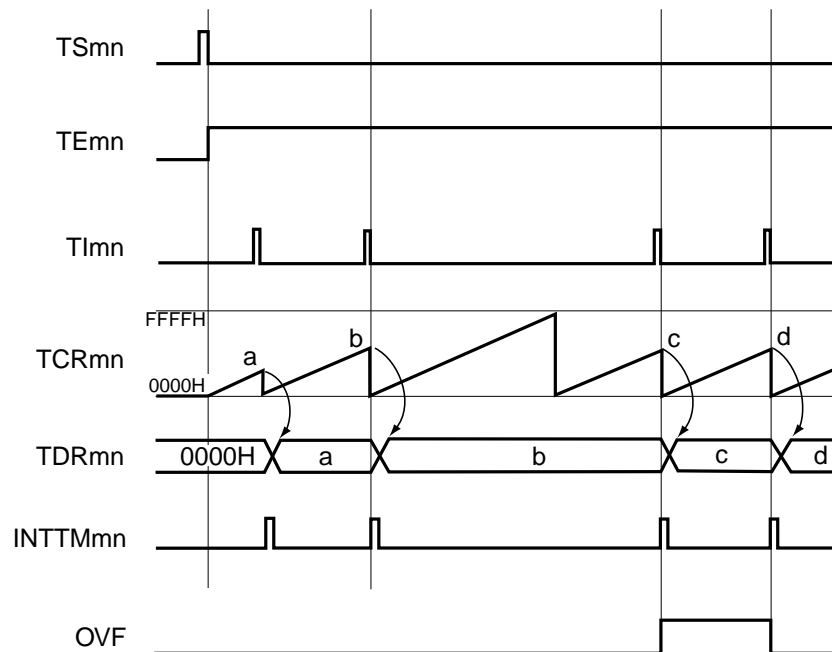
**Figure 6-51. Block Diagram of Operation as Input Pulse Interval Measurement**



**Note** When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

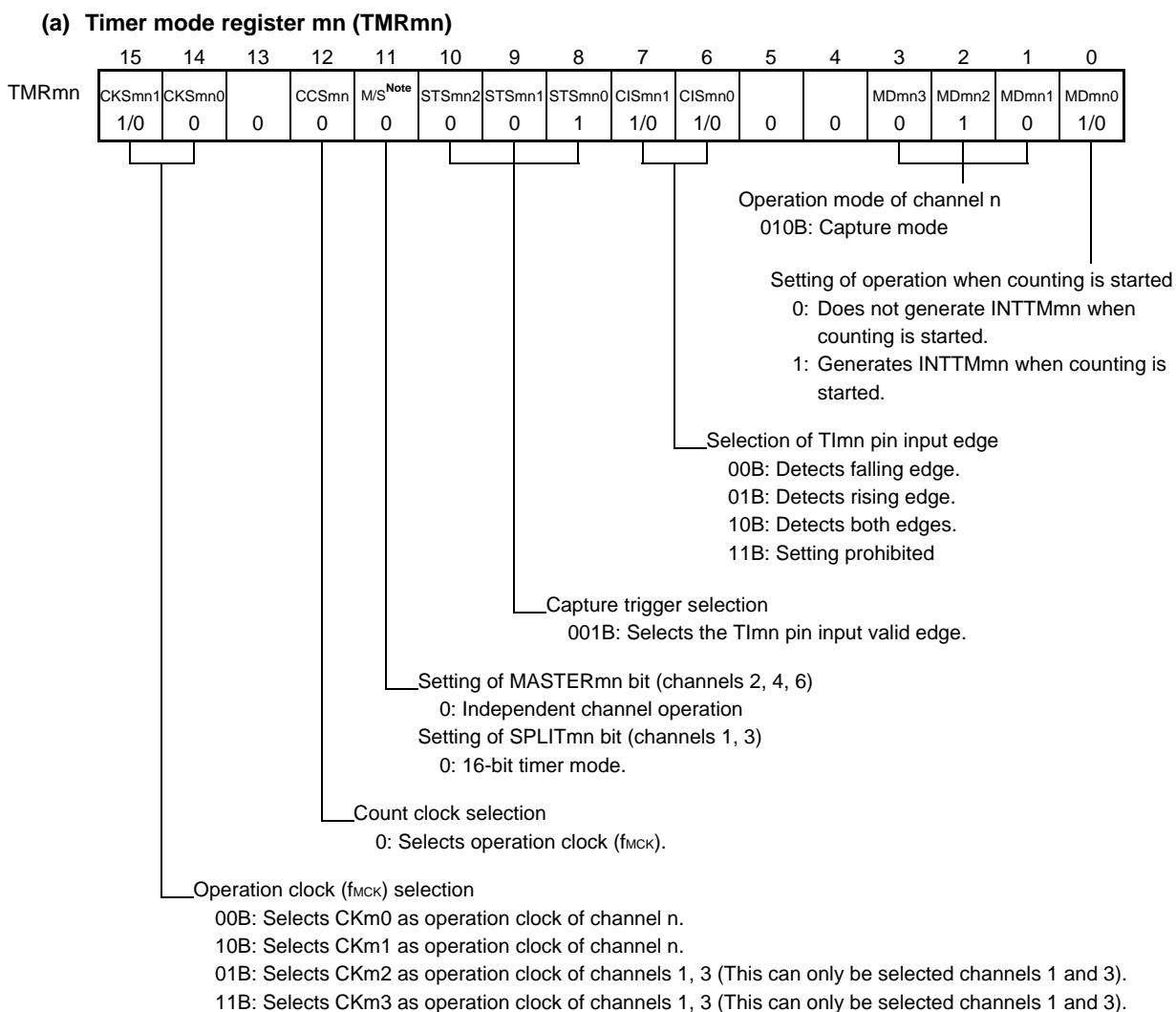
**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-52. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)

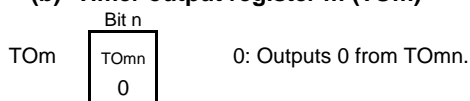


- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0 to 7)
  - TSmn: Bit n of timer channel start register m (TSm)
    - TEmn: Bit n of timer channel enable status register m (TEm)
    - TImn: TImn pin input signal
    - TCRmn: Timer count register mn (TCRmn)
    - TDRmn: Timer data register mn (TDRmn)
    - OVF: Bit 0 of timer status register mn (TSRmn)

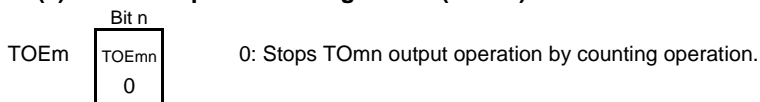
Figure 6-53. Example of Set Contents of Registers to Measure Input Pulse Interval



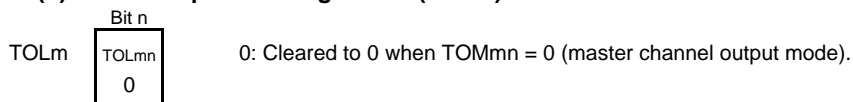
(b) Timer output register m (TOM)



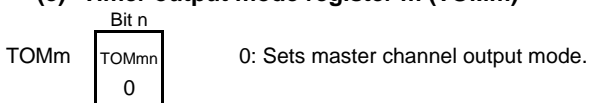
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



**Note** TMRm2, TMRm4, TMRm6: MASTERmn bit  
 TMRm1, TMRm3: SPLITmn bit  
 TMRm0, TMRm5, TMRm7: Fixed to 0

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

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 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 register 1 (NFEN1) to 0 (off) or 1 (on). 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. The TSmn bit automatically returns to 0 because it is a trigger bit.	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. 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 7)

#### 6.8.4 Operation as input signal high-/low-level width measurement

**Caution** When using a channel to implement the LIN-bus, set bit 1 (ISC1) of the input switch control register (ISC) to 1. In the following descriptions, read TImn as RxD0.

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.

$$\text{Signal width of TImn input} = \text{Period of count clock} \times ((10000\text{H} \times \text{TSRmn: OVF}) + (\text{Capture value of TDRmn} + 1))$$

**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.

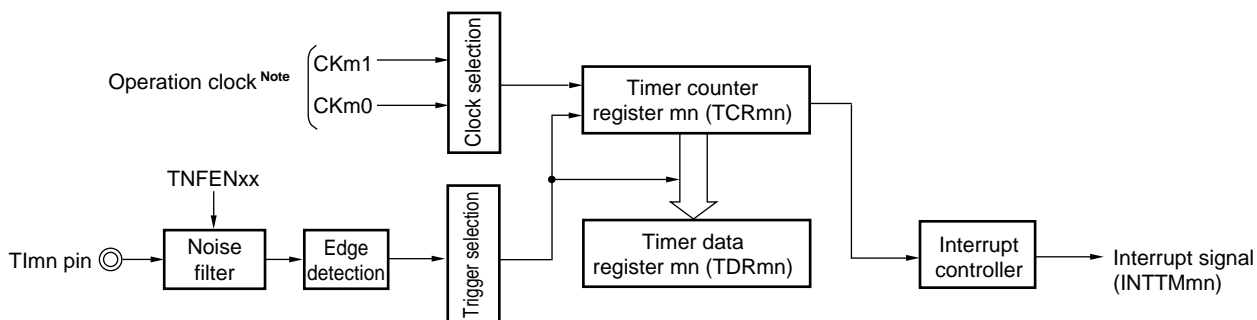
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.

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.

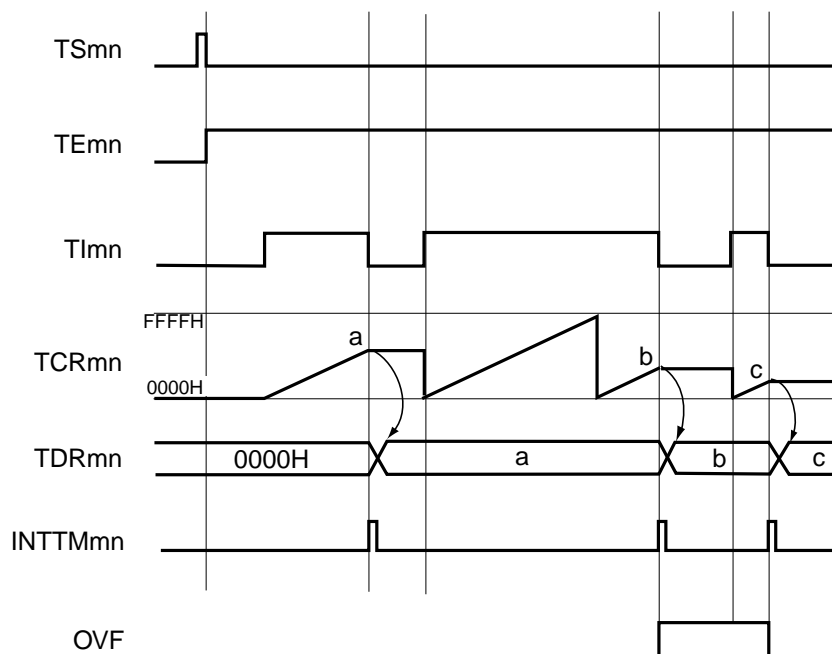
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

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.

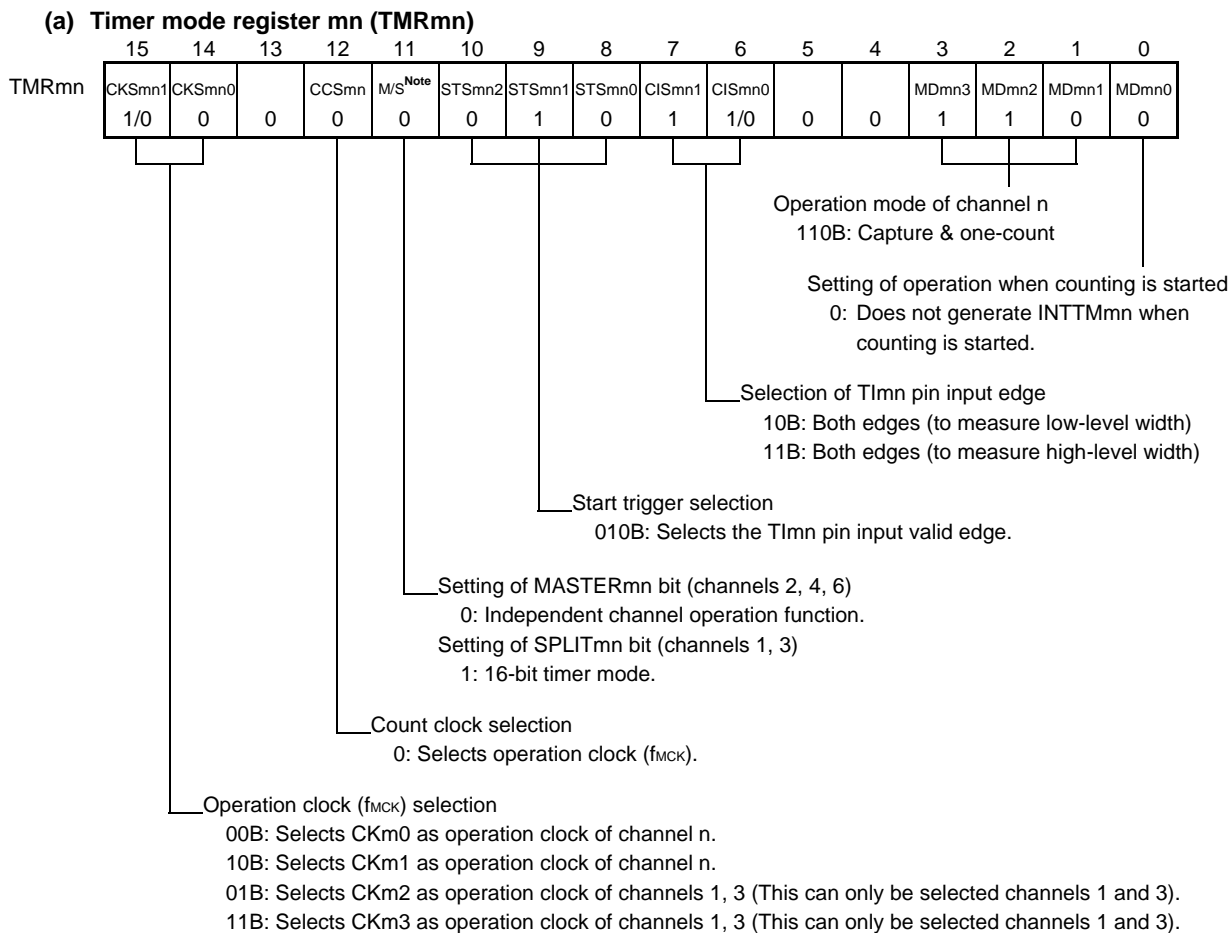
Figure 6-56. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



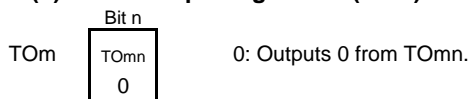
- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0 to 7)
  - TSmn: Bit n of timer channel start register m (TSM)
  - TE mn: Bit n of timer channel enable status register m (TEM)
  - TImn: TImn pin input signal
  - TCRmn: Timer count register mn (TCRmn)
  - TDRmn: Timer data register mn (TDRmn)
  - OVF: Bit 0 of timer status register mn (TSRmn)



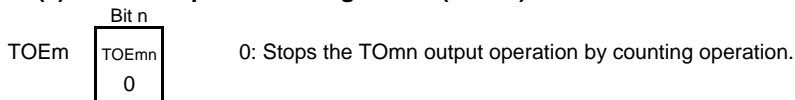
Figure 6-57. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width



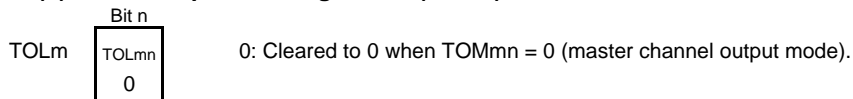
(b) Timer output register m (TOM)



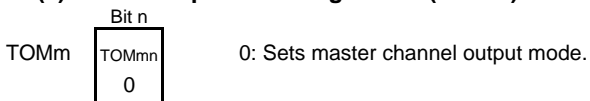
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



**Note** TMRm2, TMRm4, TMRm6: MASTERmn bit  
 TMRm1, TMRm3: SPLITmn bit  
 TMRm0, TMRm5, TMRm7: Fixed to 0

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-58. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software operation	Hardware status
<R>	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 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.)
<R>	Operation start	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.
Operation is resumed.	Set value of 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.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

### 6.8.5 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.

$$\text{Generation period of INTTMmn (timer interrupt)} = \text{Period of count clock} \times (\text{Set value of TDRmn} + 1)$$

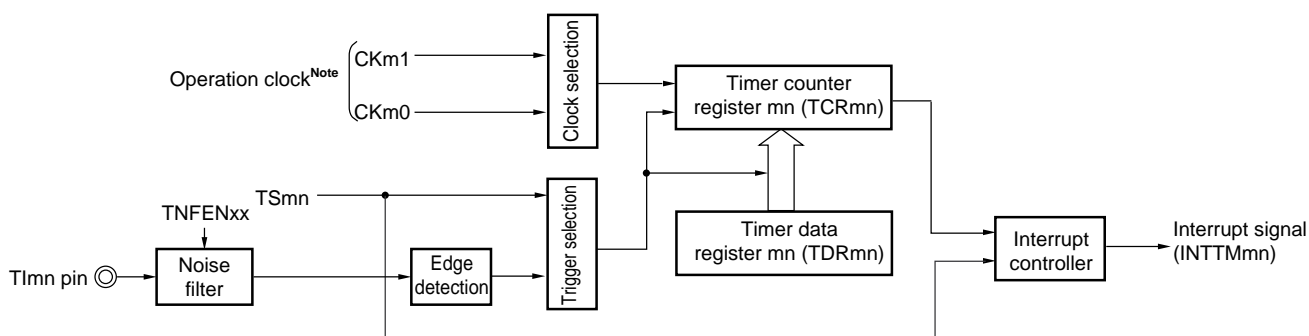
Timer count register mn (TCRmn) operates as a down counter in the one-count mode.

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.

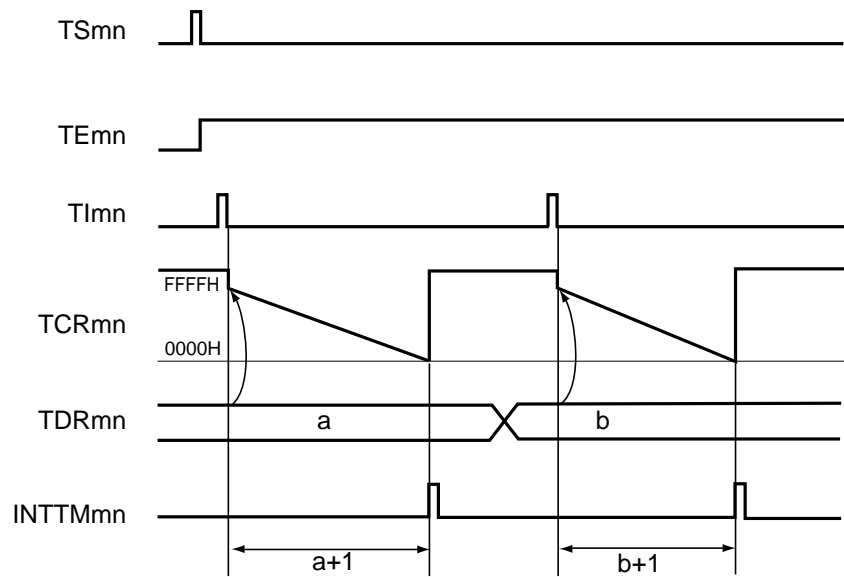
Figure 6-59. Block Diagram of Operation as Delay Counter



**Note** For using channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

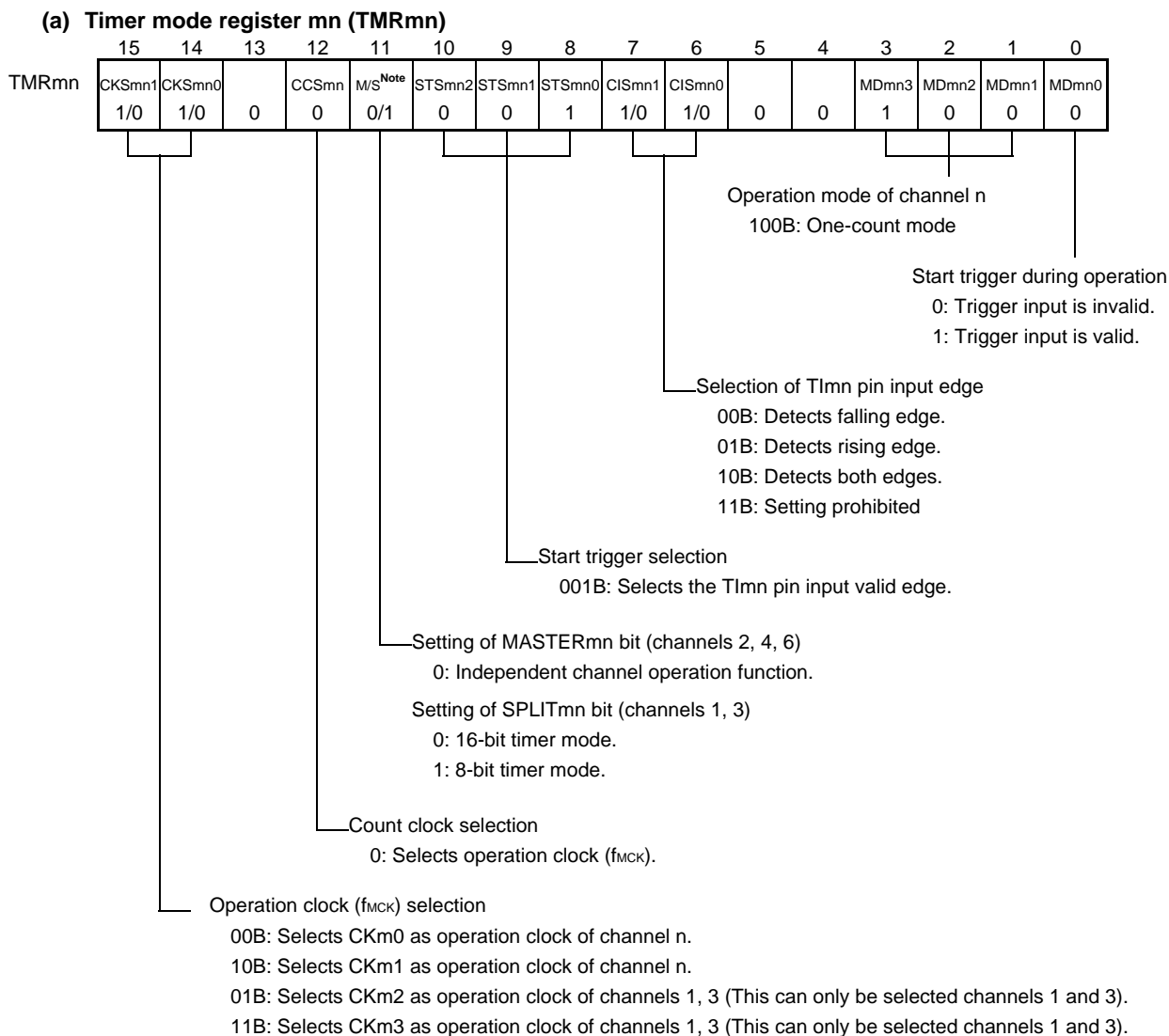
**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-60. Example of Basic Timing of Operation as Delay Counter

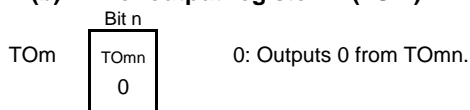


- Remarks 1.** m: Unit number (m = 0), n: Channel number (n = 0 to 7)
2. TSmn: Bit n of timer channel start register m (TSm)
  - TE mn: Bit n of timer channel enable status register m (TEm)
  - TImn: TImn pin input signal
  - TCRmn: Timer count register mn (TCRmn)
  - TDRmn: Timer data register mn (TDRmn)

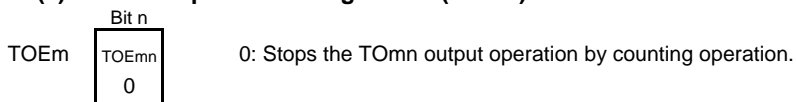
Figure 6-61. Example of Set Contents of Registers to Delay Counter (1/2)



**(b) Timer output register m (TOM)**



**(c) Timer output enable register m (TOEm)**



**Note** TMRm2, TMRm4, TMRm6: MASTERmn bit  
 TMRm1, TMRm3: SPLITmn bit  
 TMRm0, TMRm5, TMRm7: Fixed to 0

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-61. Example of Set Contents of Registers to Delay Counter (2/2)

**(d) Timer output level register m (TOLm)**

TOLm 

Bit n
TOLmn
0

 0: Cleared to 0 when TOMmn = 0 (master channel output mode).

**(e) Timer output mode register m (TOMm)**

TOMm 

Bit n
TOMmn
0

 0: Sets master channel output mode.

**Remark** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

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 register 1 (NFEN1) 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.
	Detects the TImn pin input valid edge.	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 TCRmn counts down to 0000H, INTTMmn is output, and counting stops (which leaves TCRmn at 0000H) until the next TImn pin input.
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 7)

## 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.

$\text{Delay time} = \{\text{Set value of TDRmn (master)} + 2\} \times \text{Count clock period}$ $\text{Pulse width} = \{\text{Set value of TDRmp (slave)}\} \times \text{Count clock period}$
---

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.

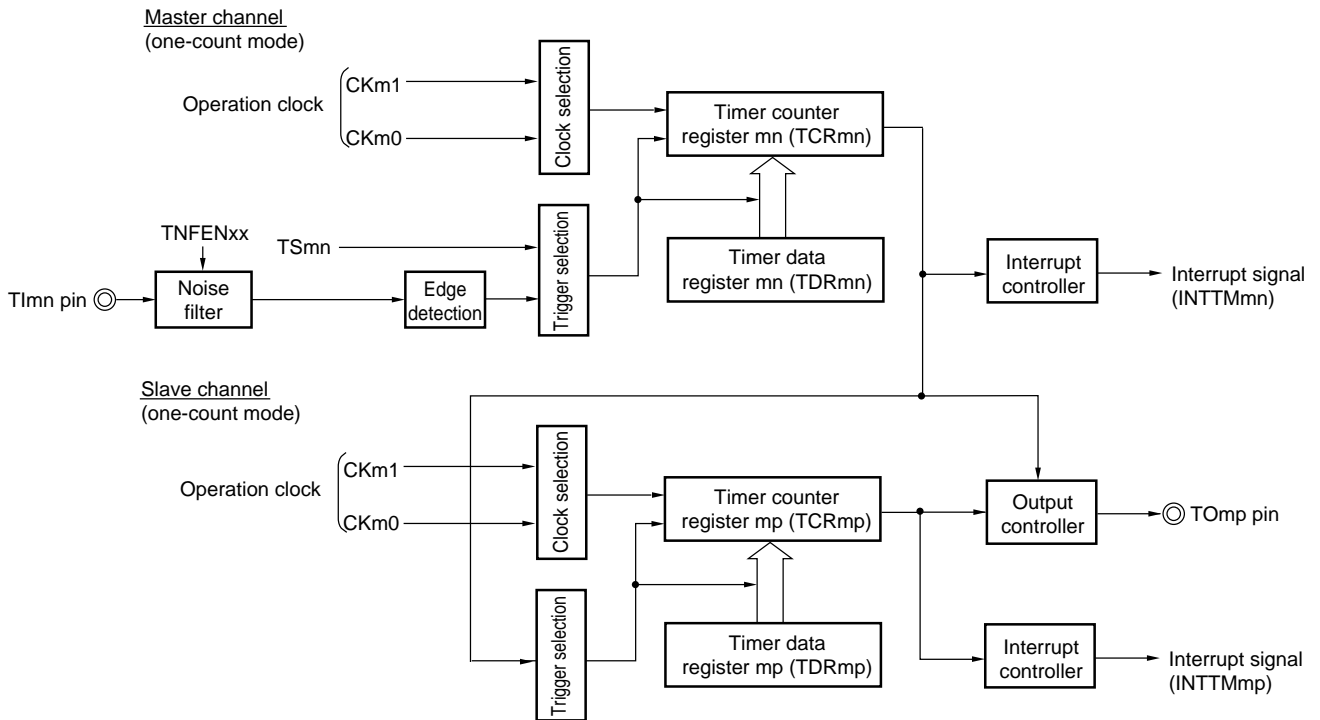
Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

<R> **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 counting, 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, 4, 6)  
p: Slave channel number (n < p ≤ 7)

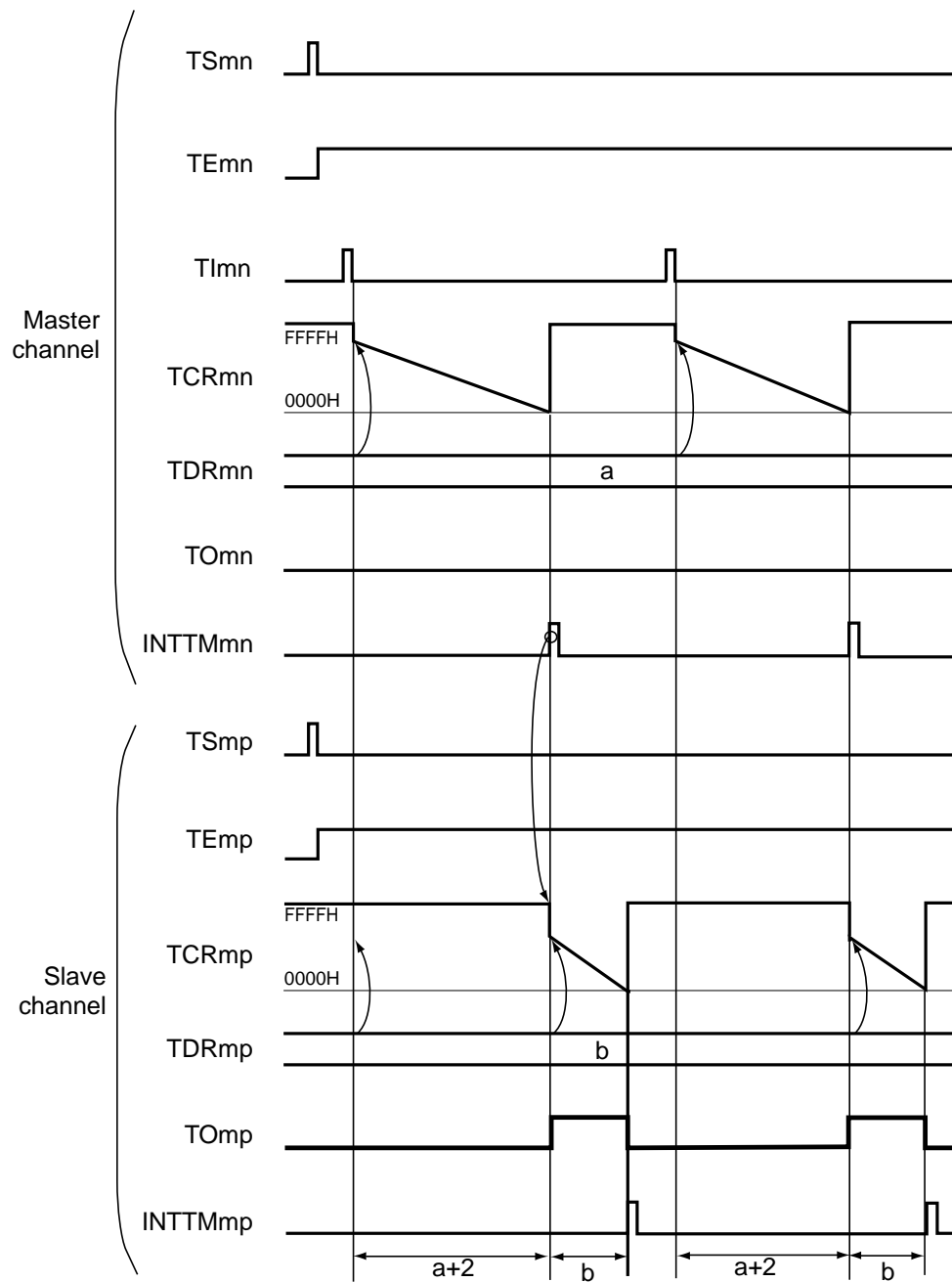


**Figure 6-63. Block Diagram of Operation as One-Shot Pulse Output Function**



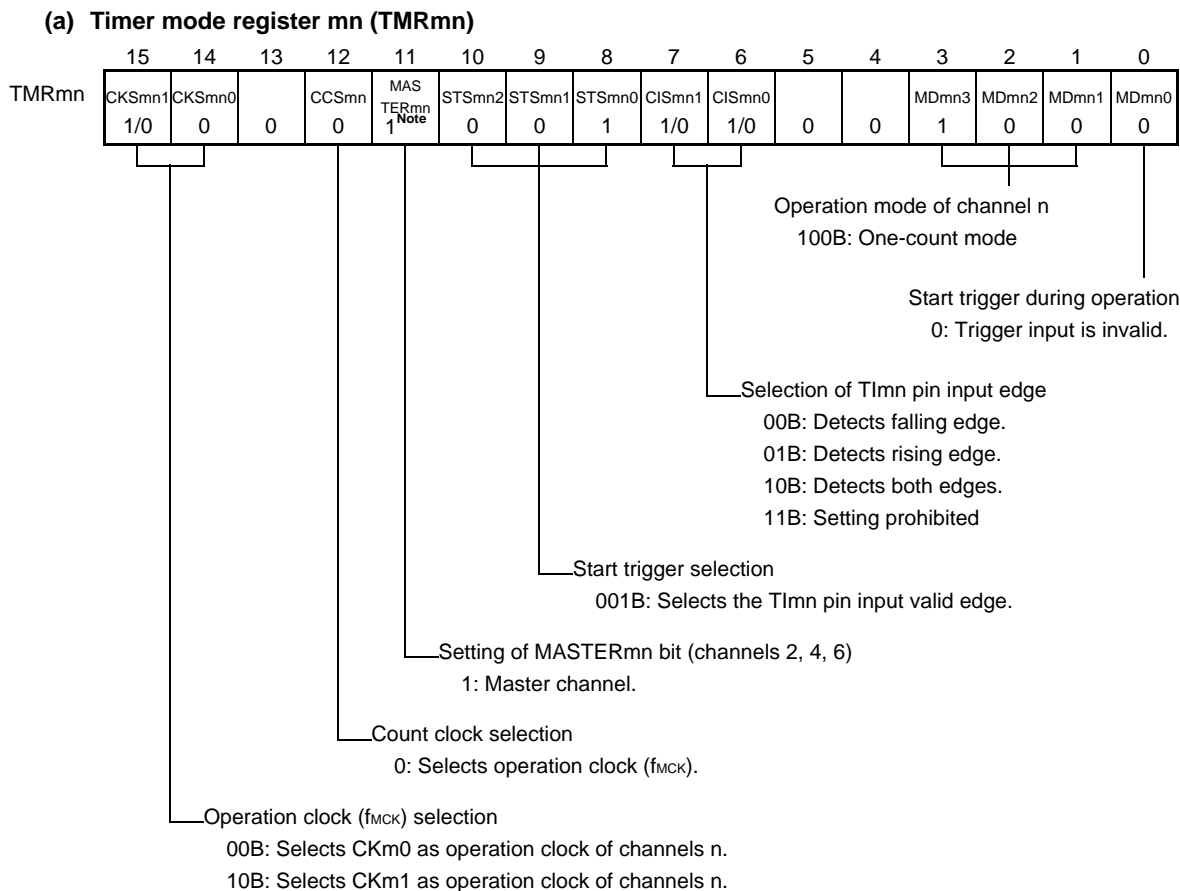
**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)  
 p: Slave channel number (n < p ≤ 7)

Figure 6-64. Example of Basic Timing of Operation as One-Shot Pulse Output Function

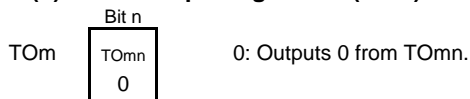


- Remarks**
1. m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)  
 p: Slave channel number (n < p ≤ 7)
  2. TS<sub>mn</sub>, TS<sub>mp</sub>: Bit n, p of timer channel start register m (TS<sub>m</sub>)  
 TE<sub>mn</sub>, TE<sub>mp</sub>: Bit n, p of timer channel enable status register m (TE<sub>m</sub>)  
 TI<sub>mn</sub>, TI<sub>mp</sub>: TI<sub>mn</sub> and TI<sub>mp</sub> pins input signal  
 TCR<sub>mn</sub>, TCR<sub>mp</sub>: Timer count registers mn, mp (TCR<sub>mn</sub>, TCR<sub>mp</sub>)  
 TDR<sub>mn</sub>, TDR<sub>mp</sub>: Timer data registers mn, mp (TDR<sub>mn</sub>, TDR<sub>mp</sub>)  
 TO<sub>mn</sub>, TO<sub>mp</sub>: TO<sub>mn</sub> and TO<sub>mp</sub> pins output signal

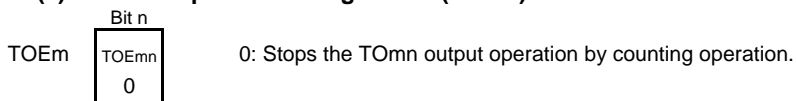
Figure 6-65. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)



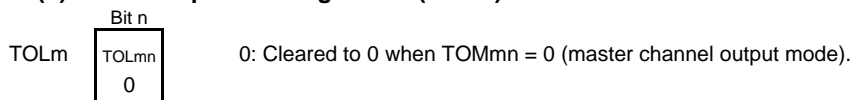
(b) Timer output register m (TOM)



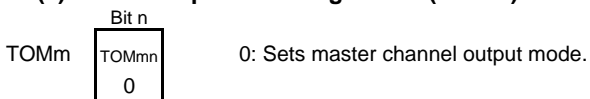
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



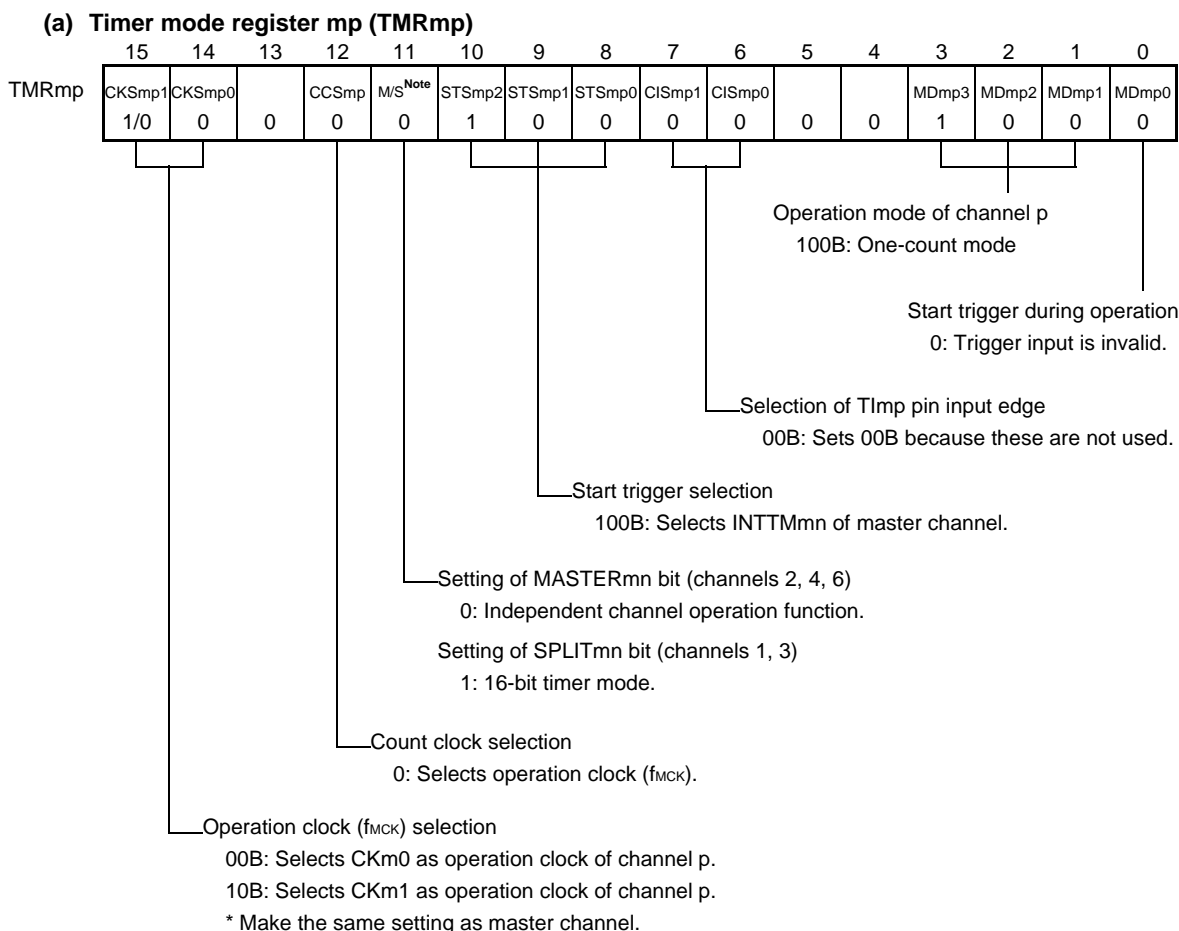
(e) Timer output mode register m (TOMm)



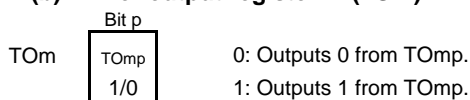
**Note** TMRm2, TMRm4, TMRm6: MASTERmn = 1  
 TMRm0: Fixed to 0

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)

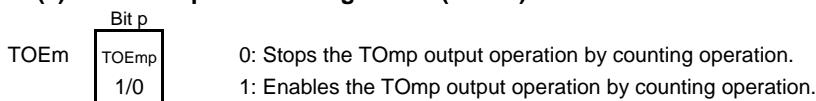
Figure 6-66. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)



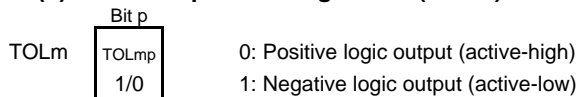
(b) **Timer output register m (TOM)**



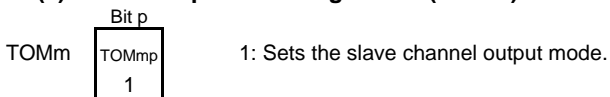
(c) **Timer output enable register m (TOEm)**



(d) **Timer output level register m (TOLm)**



(e) **Timer output mode register m (TOMm)**



**Note** TMRm2, TMRm4, TMRm6: MASTERmn bit

TMRm1, TMRm3: SPLITmp bit

TMRm5, TMRm7: Fixed to 0

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)

p: Slave channel number (n < p ≤ 7)

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 0 (off) or 1 (on). 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. Sets the TOEmp bit to 1 and enables operation of TOmp. Clears the port register and port mode register to 0.	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. TOmp does not change because channel stops operating. The TOmp pin outputs the TOmp set level.

(Remark is listed on the next page.)

Figure 6-67. Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software operation	Hardware status
Operation start	<p>Sets the TOEmp bit (slave) to 1 (only when operation is resumed).</p> <p>The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSM) are set to 1 at the same time.</p> <p>The TSmn and TSmp bits automatically return to 0 because they are trigger bits.</p>	<p>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.</p> <p>Counter stops operating.</p>
	<p>Count operation of the master channel is started by start trigger detection of the master channel.</p> <ul style="list-style-type: none"> <li>• Detects the TImn pin input valid edge.</li> <li>• Sets the TSmn bit of the master channel to 1 by software<sup>Note</sup>.</li> </ul> <p><b>Note</b> Do not set the TSmn bit of the slave channel to 1.</p>	<p>Master channel starts counting.</p>
During operation	<p>Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed.</p> <p>Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed.</p> <p>The TCRmn and TCRmp registers can always be read.</p> <p>The TSRmn and TSRmp registers are not used.</p> <p>Set values of the TOm and TOEm registers by slave channel can be changed.</p>	<p>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 valid edge is input to the TImn pin.</p> <p>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.</p> <p>After that, the above operation is repeated.</p>
Operation stop	<p>The TTmn (master) and TTmp (slave) bits are set to 1 at the same time.</p> <p>The TTmn and TTmp bits automatically return to 0 because they are trigger bits.</p>	<p>TEmn, TEmP = 0, and count operation stops.</p> <p>The TCRmn and TCRmp registers hold count value and stop.</p> <p>The TOmp output is not initialized but holds current status.</p>
	<p>The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit.</p>	<p>The TOmp pin outputs the TOmp set level.</p>
TAU stop	<p>To hold the TOmp pin output level</p> <p>Clears the TOmp bit to 0 after the value to be held is set to the port register.</p> <p>When holding the TOmp pin output level is not necessary</p> <p>Setting not required.</p>	<p>The TOmp pin output level is held by port function.</p>
	<p>The TAUmEN bit of the PER0 register is cleared to 0.</p>	<p>Power-off status</p> <p>All circuits are initialized and SFR of each channel is also initialized.</p> <p>(The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)</p>

Operation is resumed.

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)  
 p: Slave channel number (n < p ≤ 7)

### 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.

$$\begin{aligned} \text{Pulse period} &= \{\text{Set value of TDRmn (master)} + 1\} \times \text{Count clock period} \\ \text{Duty factor [\%]} &= \{\text{Set value of TDRmp (slave)}\} / \{\text{Set value of TDRmn (master)} + 1\} \times 100 \\ \text{0\% output:} & \quad \text{Set value of TDRmp (slave)} = 0000\text{H} \\ \text{100\% output:} & \quad \text{Set value of TDRmp (slave)} \geq \{\text{Set value of TDRmn (master)} + 1\} \end{aligned}$$

**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.

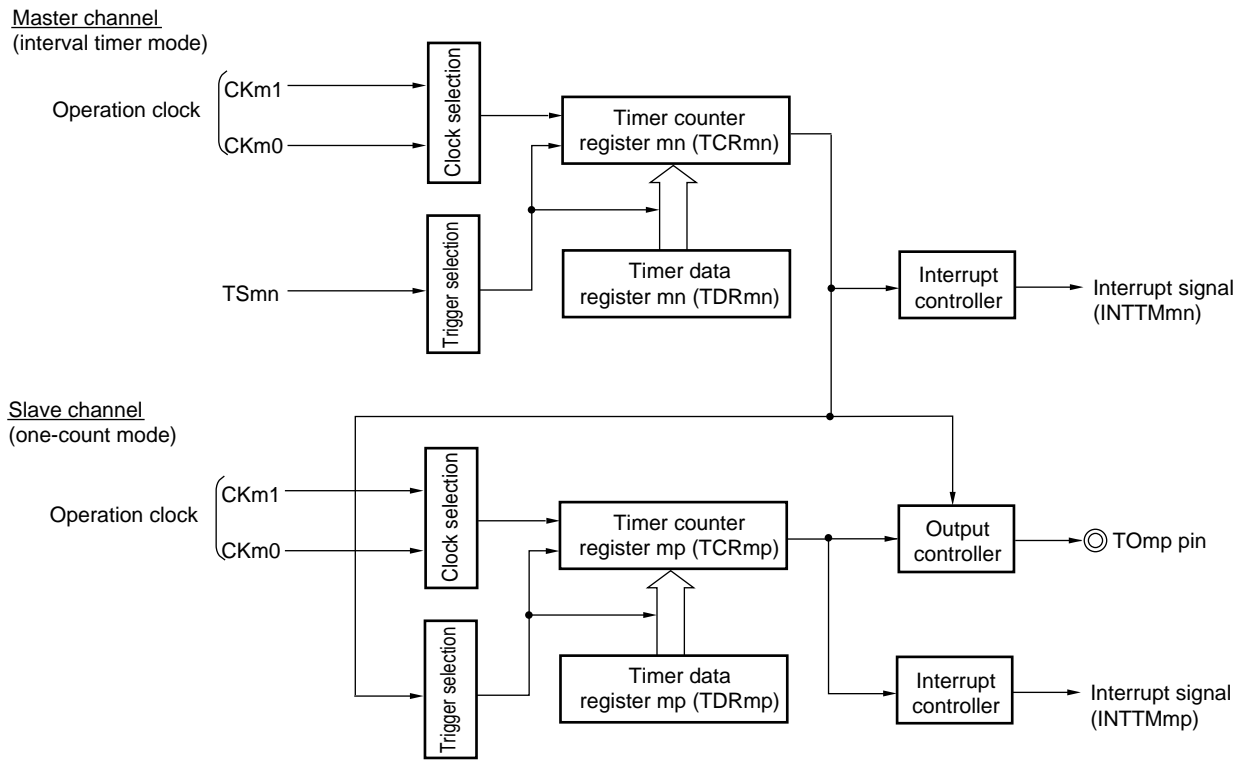
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.

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, 4, 6)  
p: Slave channel number (n < p ≤ 7)

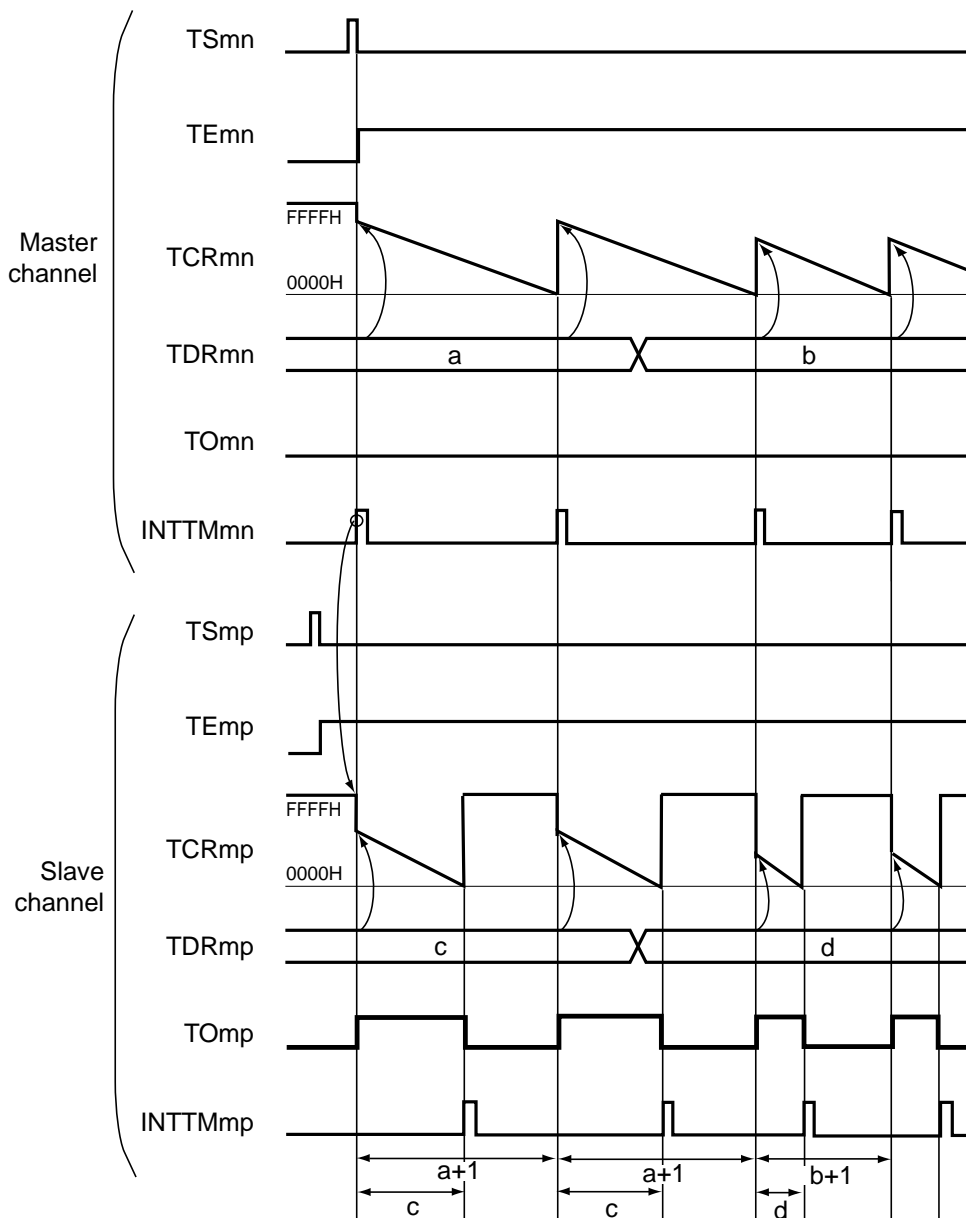
Figure 6-68. Block Diagram of Operation as PWM Function



**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)  
 p: Slave channel number (n < p ≤ 7)

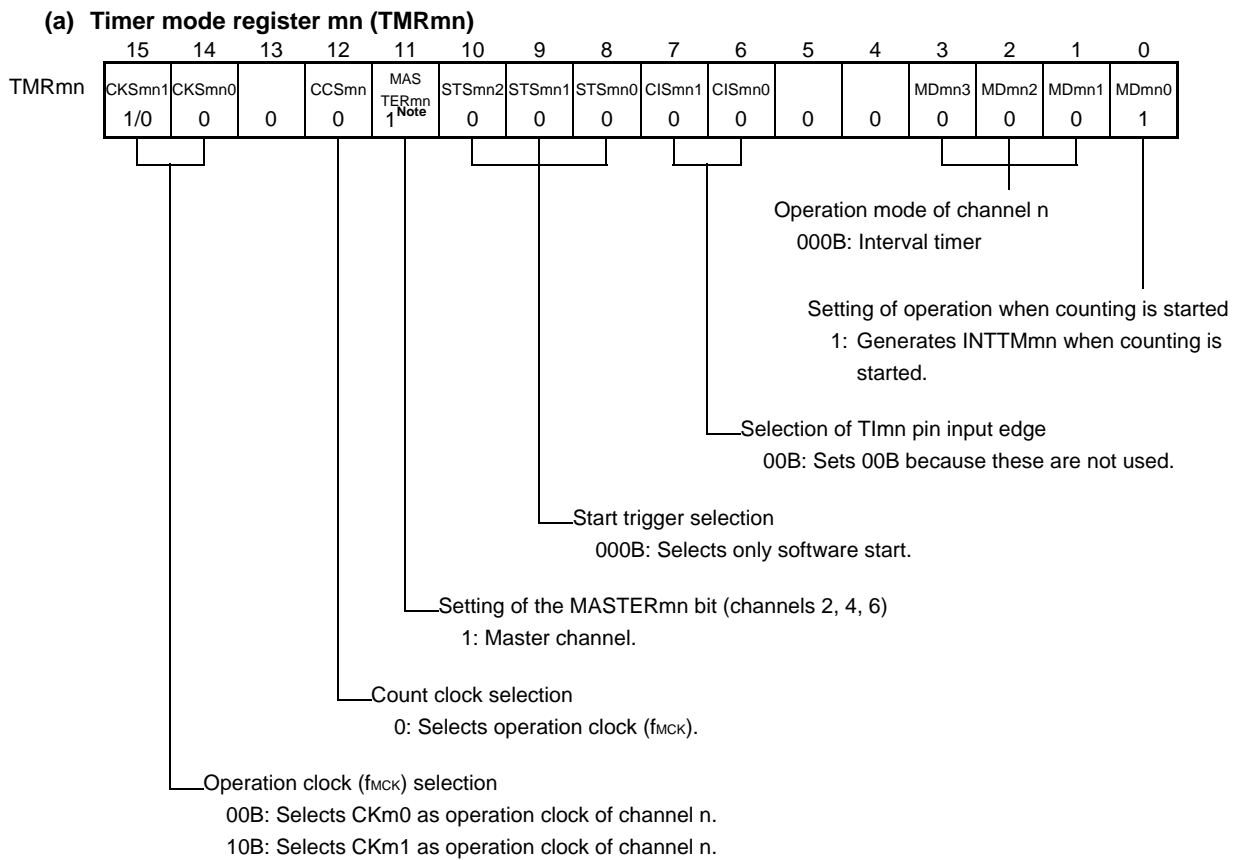


Figure 6-69. Example of Basic Timing of Operation as PWM Function

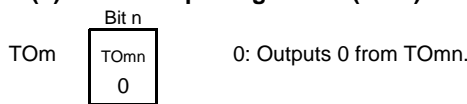


- Remarks 1.** m: Unit number ( $m = 0$ ), n: Channel number ( $n = 0, 2, 4, 6$ )  
 p: Slave channel number ( $n < p \leq 7$ )
- 2.**  $TS_{mn}$ ,  $TS_{mp}$ : Bit n, p of timer channel start register m ( $TS_m$ )  
 $TE_{mn}$ ,  $TE_{mp}$ : Bit n, p of timer channel enable status register m ( $TE_m$ )  
 $TCR_{mn}$ ,  $TCR_{mp}$ : Timer count registers mn, mp ( $TCR_{mn}$ ,  $TCR_{mp}$ )  
 $TDR_{mn}$ ,  $TDR_{mp}$ : Timer data registers mn, mp ( $TDR_{mn}$ ,  $TDR_{mp}$ )  
 $TO_{mn}$ ,  $TO_{mp}$ :  $TO_{mn}$  and  $TO_{mp}$  pins output signal

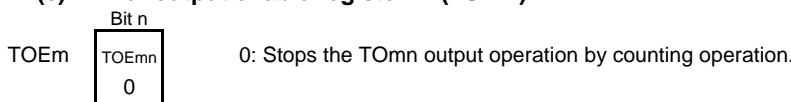
Figure 6-70. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used



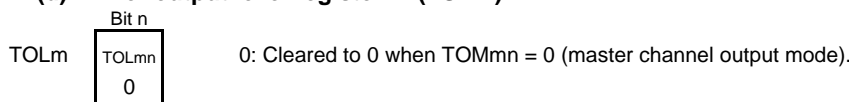
(b) Timer output register m (TOM)



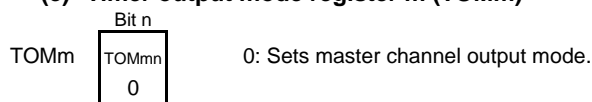
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)

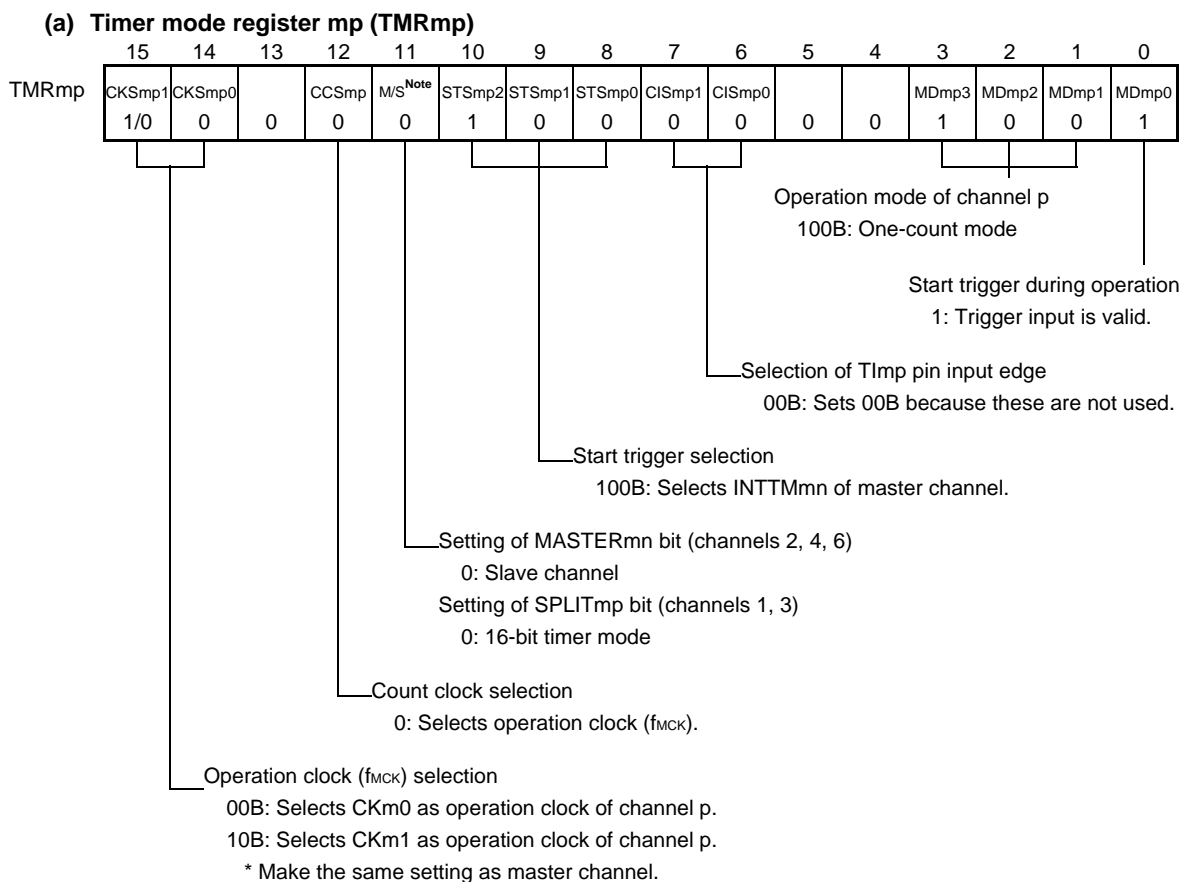


**Note** TMRm2, TMRm4, TMRm6: MASTERmn = 1

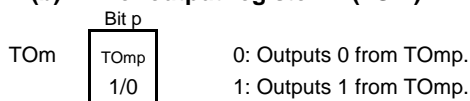
TMRm0: Fixed to 0

**Remark**m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)

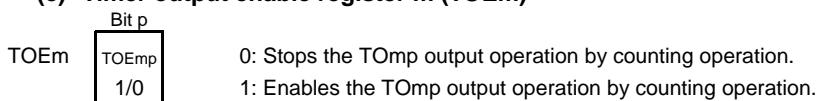
Figure 6-71. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used



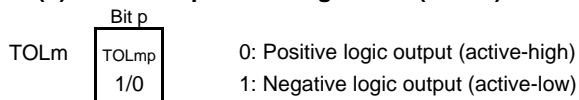
(b) Timer output register m (TOM)



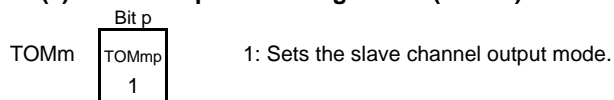
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



**Note** TMRm2, TMRm4, TMRm6: MASTERmn bit  
 TMRm1, TMRm3: SPLITmp bit  
 TMRm5, TMRm7: Fixed to 0

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)  
 p: Slave channel number (n < p ≤ 7)

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 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 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. Sets the TOEmp bit to 1 and enables operation of TOmp. Clears the port register and port mode register to 0.	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. TOmp does not change because channel stops operating. The TOmp pin outputs the TOmp set level.

(Remark is listed on the next page.)

Figure 6-72. Operation Procedure When PWM Function Is Used (2/2)

	Software operation	Hardware status
Operation is resumed.	<b>Operation start</b> 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.
	<b>During operation</b> 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.
	<b>Operation stop</b> 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.
	<b>TAU stop</b> 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, 4, 6)  
 p: Slave channel number (n < p ≤ 7)

### 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.

$$\begin{aligned} \text{Pulse period} &= \{\text{Set value of TDRmn (master)} + 1\} \times \text{Count clock period} \\ \text{Duty factor 1 [\%]} &= \{\text{Set value of TDRmp (slave 1)}\} / \{\text{Set value of TDRmn (master)} + 1\} \times 100 \\ \text{Duty factor 2 [\%]} &= \{\text{Set value of TDRmq (slave 2)}\} / \{\text{Set value of TDRmn (master)} + 1\} \times 100 \end{aligned}$$

**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.

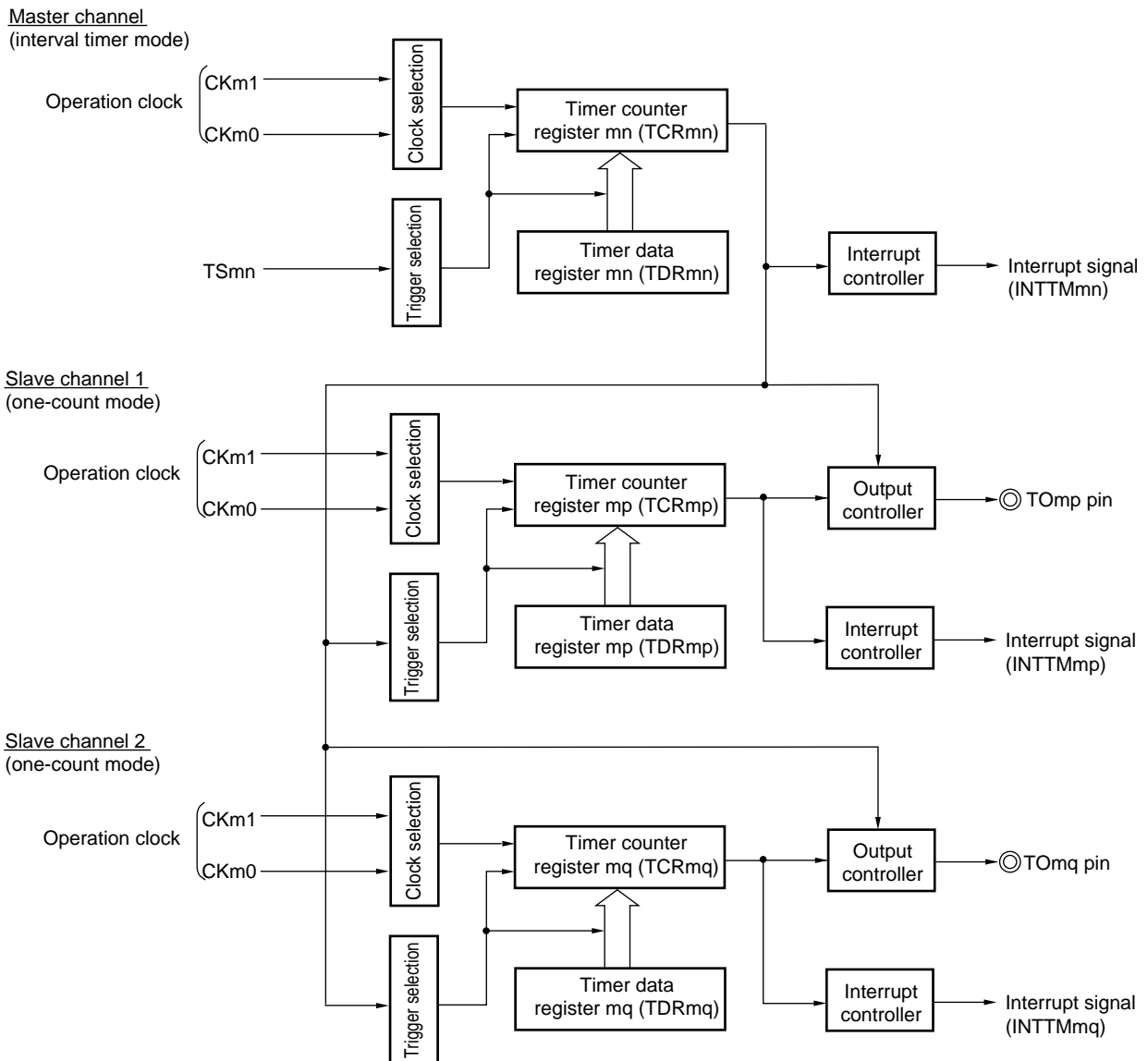
In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOMq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a 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 seven 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).

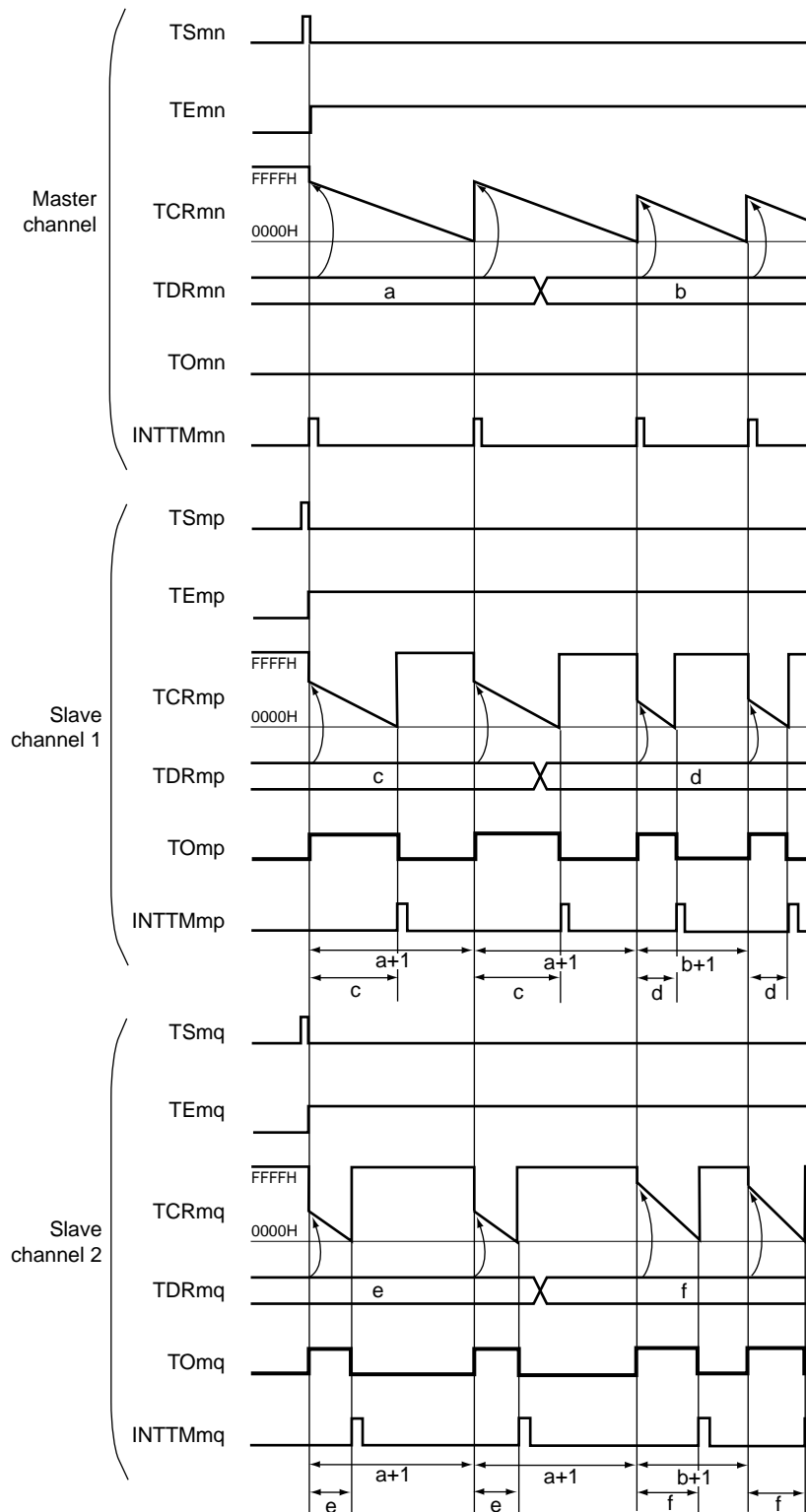
**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)  
p: Slave channel number 1, q: Slave channel number 2  
n < p < q ≤ 7 (Where p and q are integers greater than n)

**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, 2, 4)  
 p: Slave channel number 1, q: Slave channel number 2  
 n < p < q ≤ 7 (Where p and q are integers greater than n)

**Figure 6-74. Example of Basic Timing of Operation as Multiple PWM Output Function (Output Two Types of PWMs)**

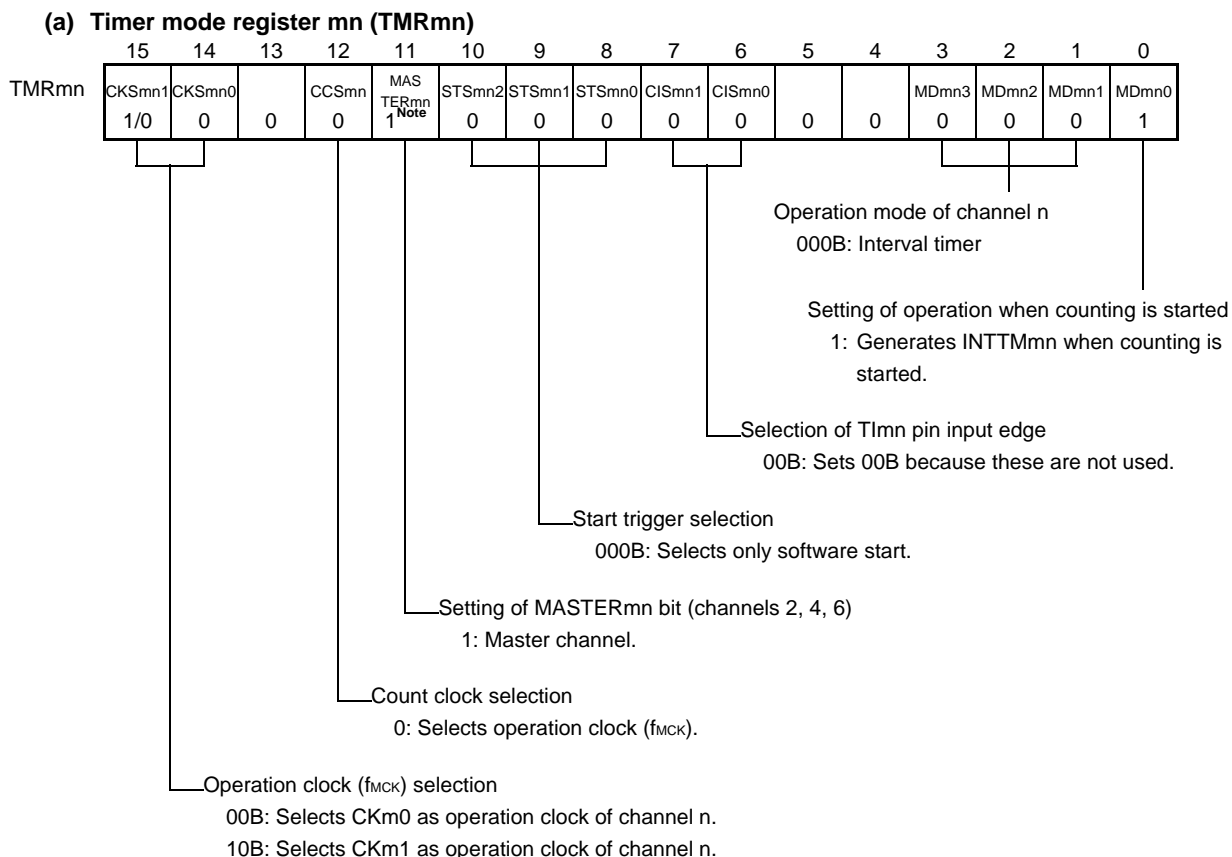


(Remark is listed on the next page.)

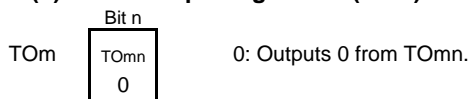


- Remarks**
1. m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)  
p: Slave channel number 1, q: Slave channel number 2  
n < p < q ≤ 7 (Where p and q are integers greater than n)
  2. TS<sub>mn</sub>, TS<sub>mp</sub>, TS<sub>mq</sub>: Bit n, p, q of timer channel start register m (TSM)  
TE<sub>mn</sub>, TE<sub>mp</sub>, TE<sub>mq</sub>: Bit n, p, q of timer channel enable status register m (TEM)  
TCR<sub>mn</sub>, TCR<sub>mp</sub>, TCR<sub>mq</sub>: Timer count registers mn, mp, mq (TCR<sub>mn</sub>, TCR<sub>mp</sub>, TCR<sub>mq</sub>)  
TDR<sub>mn</sub>, TDR<sub>mp</sub>, TDR<sub>mq</sub>: Timer data registers mn, mp, mq (TDR<sub>mn</sub>, TDR<sub>mp</sub>, TDR<sub>mq</sub>)  
TOM<sub>n</sub>, TOM<sub>p</sub>, TOM<sub>q</sub>: TOM<sub>n</sub>, TOM<sub>p</sub>, and TOM<sub>q</sub> pins output signal

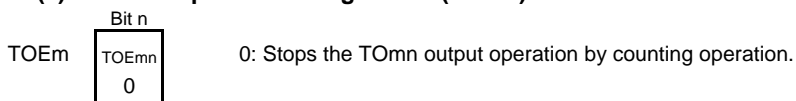
**Figure 6-75. Example of Set Contents of Registers  
When Multiple PWM Output Function (Master Channel) Is Used**



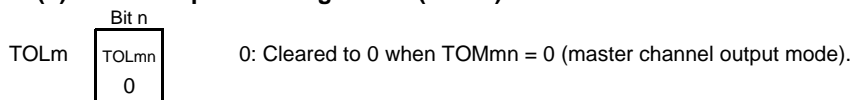
**(b) Timer output register m (TOM)**



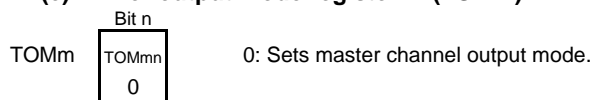
**(c) Timer output enable register m (TOEm)**



**(d) Timer output level register m (TOLm)**



**(e) Timer output mode register m (TOMm)**

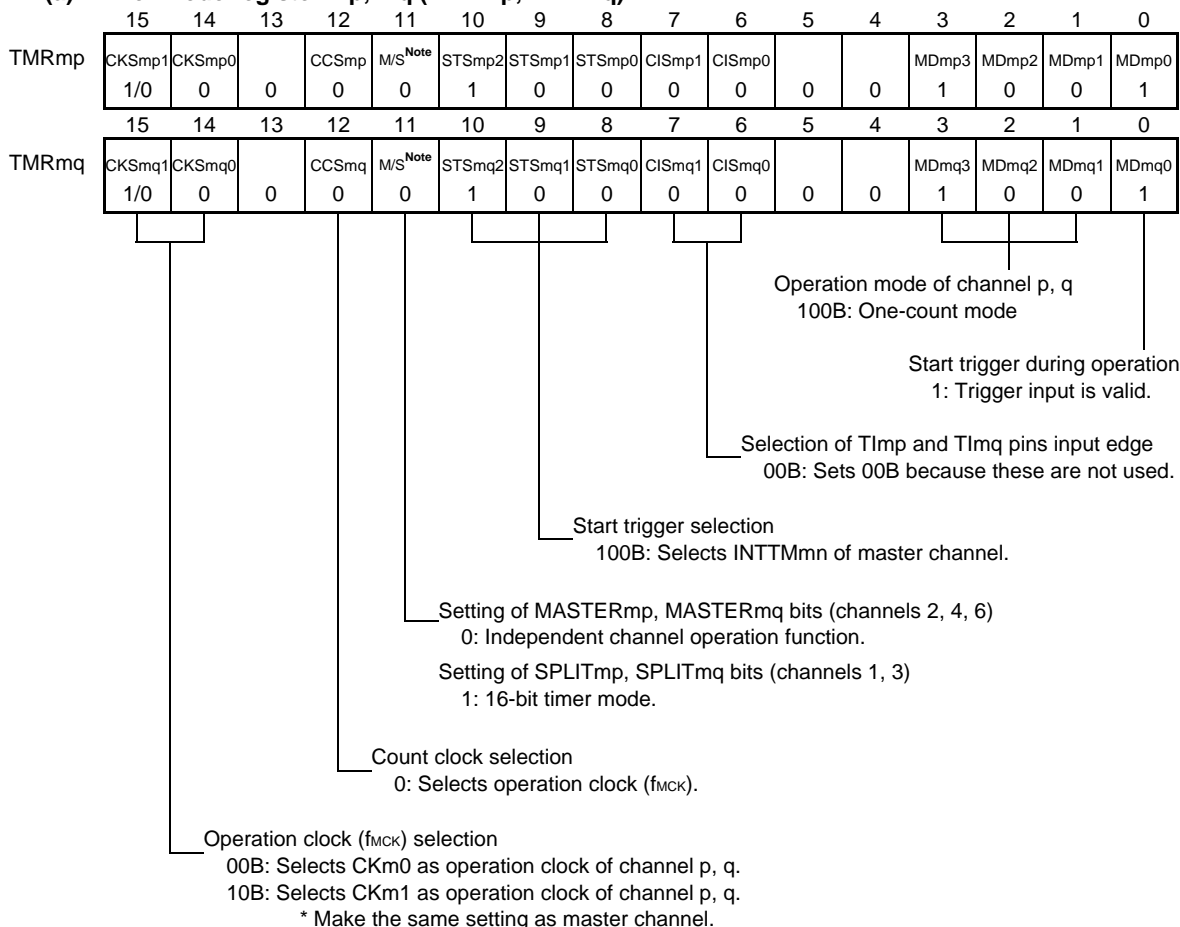


**Note** TMRm2, TMRm4, TMRm6: MASTERmn = 1  
TMRm0: Fixed to 0

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)

**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)**



**(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** TMRm2, TMRm4, TMRm6: MASTERmp, MASTERmq bit  
 TMRm1, TMRm3: SPLITmp, SPLITmq bit  
 TMRm5, TMRm7: Fixed to 0

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)  
 p: Slave channel number 1, q: Slave channel number 2  
 n < p < q ≤ 7 (Where p and q are 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 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 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). Sets the TOLmp and TOLmq bits. 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.

(Remark is listed on the next page.)

Figure 6-77. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

	Software operation	Hardware status
Operation is resumed.	<p>Operation start</p> <p>(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.</p>	<p>TEmn = 1, TEmq = 1</p> <p>When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.</p>
	<p>During operation</p> <p>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.</p>	<p>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 register, 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.</p>
	<p>Operation stop</p> <p>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.</p>	<p>TEmn, 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.</p>
	<p>The TOEmp and TOEmq bits of slave channels are cleared to 0 and value is set to the TOMp and TOMq bits.</p>	<p>The TOmp and TOMq pins output the TOmp and TOMq set levels.</p>
TAU stop	<p>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</p>	<p>The TOmp and TOMq pin output levels are held by port function.</p>
	<p>The TAUmEN bit of the PER0 register is cleared to 0.</p>	<p>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.)</p>

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)  
 p: Slave channel number 1, q: Slave channel number 2  
 n < p < q ≤ 7 (Where p and q are a consecutive integer greater than n)

**6.9.4 Remote control output function**

The PWM output function is applied to the remote control output function.

The pairings of channels 2 and 3 and channels 4 and 5 are used to output the PWM signal (See **6.9.2 Operation as PWM function** for how to set up each channel.). The PWM signal output from channel 3 is used as a mask wave, the PWM signal output from channel 5 is used as a carrier waves, and the logical products of these signals are output as remote control output.

The high level width output part of the remote control output is composed of a 20 to 60 kHz carrier signal.

**Figure 6-78. Remote Control Output**

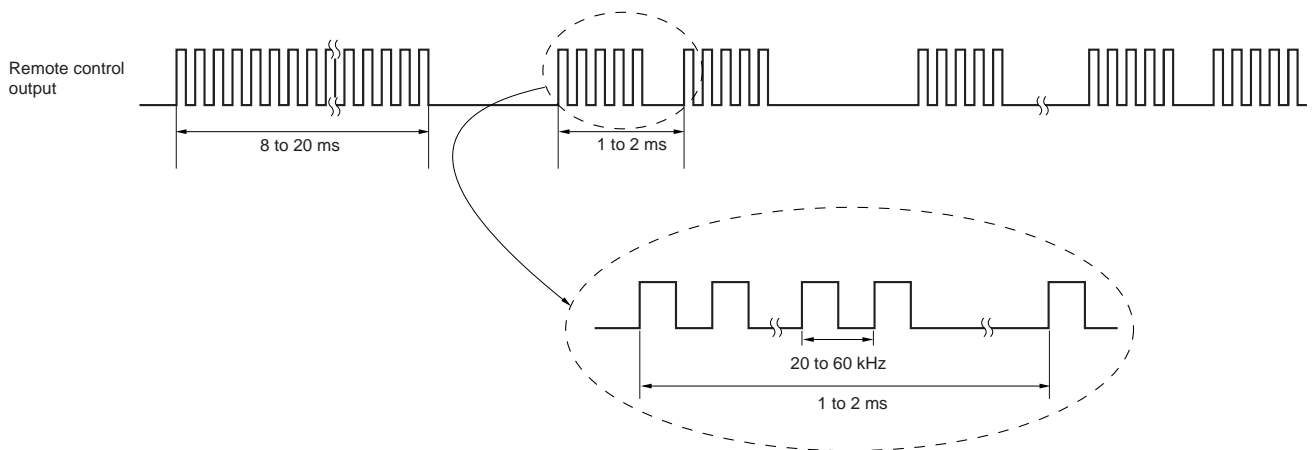
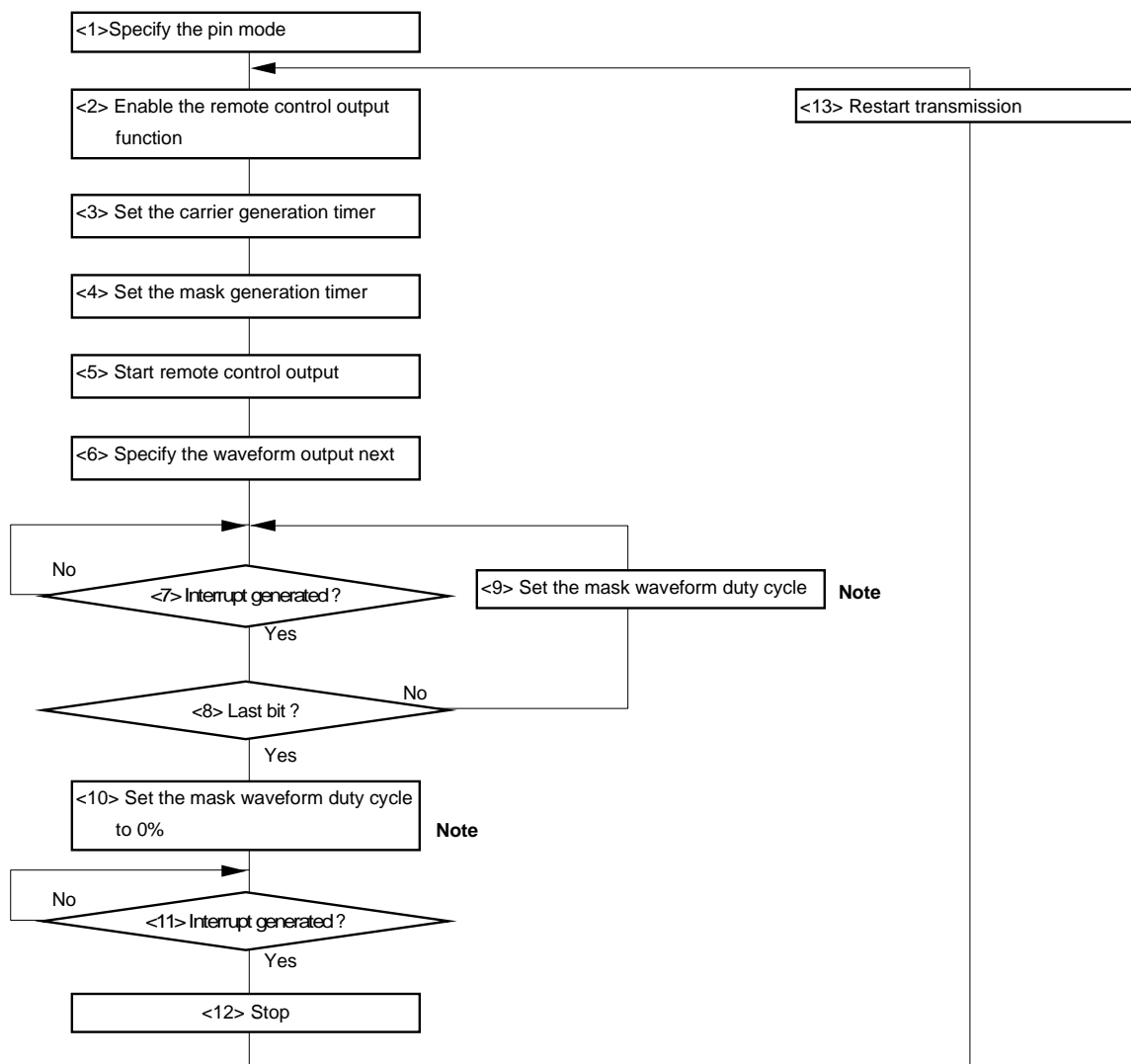


Figure 6-78 shows the steps for setting the remote control output.

Figure 6-79. Procedure for Setting Remote control Output (1/2)



<1> Specify the pin mode.

Clear the PFSEG20 bit of the PFSEG2 register to 0, the PM30 bit of the PM3 register to 0, the PU30 bit of the PU3 register to 0, and the P30 bit of the P3 register to 0.

<2> Set the TOS0 bit of the timer output select register (TOS) to 1.

<3> Specify the carrier waveform by using the PWM function for channel 4 (master) and channel 5 (slave).

TDR04 register value = Carrier waveform cycle - 1

TDR05 register value = Carrier waveform high-level width

<4> Generate the mask waveform by using the PWM function for channel 2 (master) and channel 3 (slave).

Specify the mask waveform cycle and high-level width (header code)

TDR02 register value = Mask waveform cycle - 1

TDR03 register value = Mask waveform high-level width

<5> Start output operation.

Set the TS02, TS03, TS04, and TS05 bits of timer channel start register 0 (TS0) to 1 at the same time.

**Figure 6-79. Procedure for Setting Remote control Output (2/2)**

<6> Specify the waveform to be output next.

TDR02 register value = Mask waveform cycle - 1

TDR03 register value = Mask waveform high-level width

<7> Wait for an interrupt signal (INTTM02) to be generated at the rise of the mask waveform.

<8> Jump to step <10> for processing that stops output if the bit is the final code bit of the remote control carrier waveform.

If not the final code bit, set the next mask waveform in step <9> and repeat steps <7> to <9> until the final code bit is received.

[Consecutive transmission]

<9> Specify the cycle and duty (cycle: TDR02; high level width: TDR03).

[Stopping remote control output]

<10> Set the mask waveform duty to 0%. (Set TDR03 to 0000H.)

<11> Wait for an interrupt signal (INTTM02) to be generated at the rise of the mask waveform.

<12> Stop the timer.

Set the TT02, TT03, TT04, and TT05 bits of timer channel stop register 0 (TT0) to 1 at the same time, clear the TOE02, TOE03, TOE04, and TOE05 bits of timer output enable register 0 (TOE0) to 0, and then clear the TO02, TO03, TO04, and TO05 bits of timer output register 0 (TO0) to 0.

[Restarting remote control output]

<13> To restart transmission, set the TOE03 and TOE05 bits to 1 and re-specify the settings from step <2>. (It is not necessary to overwrite the same value.)

**Note** Setting values are applied at the rise of the mask waveform.

The mask waveform cycle and high level width can only be specified as an integral multiple of the carrier cycle.

- Cautions**
1. During the period between <3> and <11>, do not stop supplying a clock to TAU (by using a STOP instruction, etc.) or change the value of registers other than TDR02 and TDR03.
  2. The system must be in the normal operation mode or the HALT mode during steps <3> to <11>.
  3. Select the same operation clock for channels 2, 3, 4, and 5.
  4. Be sure to perform steps <7> to <9> to stop remote control output.  
The following errors might occur if stopped using a different method:
    - A waveform with a cycle that is not the same as the carrier is output.
    - The timer output is fixed to a high level after the operation stops.
  5. When performing remote control output, be sure to set channels 2, 3, 4, and 5 to PWM output mode.



## 6.10 Cautions When Using Timer Array Unit

### 6.10.1 Cautions when using timer output

Depends on products, a pin is assigned a timer output and other alternate functions. In this case, outputs of the other alternate functions must be set in initial status.

#### (1) 80-pin products

##### (a) Using TO00 to TO07 outputs

In addition to clearing the port mode register (the PMxx bit) and the port register (the Pxx bit) to 0, be sure to clear the corresponding bit of LCD port function registers 0 to 6 (PFSEG0 to PFSEG6) to "0". For details, see **Table 4-6 Setting Examples of Registers and Output Latches When Using Alternate Function.**

##### (b) Using TO05 output assigned to the P42 (When PIOR2 = 1)

So that the alternated SO10/TxD1 output becomes 1, not only set the port mode register (the PM42 bit) and the port register (the P42 bit) to 0, but also use the serial channel enable status register 0 (SE0), serial output register 0 (SO0), and serial output enable register 0 (SOE0) with the same setting as the initial status.

#### (2) 64-pin products

##### (a) Using TO00 to TO05 and TO07 outputs

In addition to clearing the port mode register (the PMxx bit) and the port register (the Pxx bit) to 0, be sure to clear the corresponding bit of LCD port function registers 0 to 6 (PFSEG0 to PFSEG6) to "0". For details, see **Table 4-6 Setting Examples of Registers and Output Latches When Using Alternate Function.**

##### (b) Using TO05 output assigned to the P42 (When PIOR2 = 1)

So that the alternated SO10/TxD1 output becomes 1, not only set the port mode register (the PM42 bit) and the port register (the P42 bit) to 0, but also use the serial channel enable status register 0 (SE0), serial output register 0 (SO0), and serial output enable register 0 (SOE0) with the same setting as the initial status.

##### (c) Using TO06 output assigned to the P125

In addition to clearing the port mode register (the PM125 bit) and the port register (the P125 bit) to 0, be sure to clear the ISCVL3 bit of the LCD Input switch control register (ISCLCD) to "0". For details, see **Table 4-6 Setting Examples of Registers and Output Latches When Using Alternate Function.**

## CHAPTER 7 16-BIT TIMER KB20

<R> 16-bit timer KB20 is a timer that can generate PWM output which is suitable to control power sources and lighting. The 16-bit timer KB20 (IH) incorporated in the RL78/L13 also has the IH-only PWM output function.

	80-pin	64-pin
16-bit timer KB20 (IH)	√	√

- Cautions 1.** Most of the following descriptions in this chapter use the 80-pin products as an example.
- 2.** In this manual, the 16-bit timer KB20 (IH) incorporated in the RL78/L13 may be referred to as the 16-bit timer KB20.

### 7.1 Functions of 16-bit Timer KB20

16-bit timer KB20 is a dedicated PWM output timer and has two outputs (one of two outputs can be selected from three pins). This timer is provided with the following functions.

**(1) PWM output**

- A variable PWM with any duty or cycle can be output while the timer is operating.
- The default timer output level (high or low level) can be set.

**(2) Timer restart function**

Timer output can be restarted directly (not via the CPU) when a trigger source (counter restart trigger source 0 to 2 or IH-PWM output restart request signal) occurs.

**(3) Forced output stop function 1 (by interlocking with the comparator, INTP)**

Timer output can be fixed to high impedance, high, or low level directly (not via the CPU) when a trigger source (comparator 0, 1, INTP0, INTP1) occurs. The stop function is cancelled by the stop trigger setting of forced output stop function 1.

**(4) Forced output stop function 2 (by interlocking with the comparator, INTP)**

Timer output can be fixed to high or low level directly (not via the CPU) when a trigger source occurs (comparator 0, 1, INTP0, INTP1, INTP2). When the following counter period is started or when a trigger source is eliminated, the stop function is cancelled directly (not via the CPU).

**(5) Dithering function**

The “set duty + 1” waveform in each 16-period cycle can be output in the range of periods 0 to 15.

**(6) Smooth start function**

It is possible to make a smooth start that automatically increases the duty after PWM output starts until it reaches the configured duty value.

It is possible to configure the initial duty and duty plus one incremental period.

**(7) Maximum frequency setting function**

With the timer restart function, restart can be held pending until the set period.

**(8) Interleave function**

With the timer restart function, it is possible to use external sources to automatically alternate restart output between two outputs. It is possible to make interleaved PFC control with critical conduction mode.

**Remark** Critical conduction mode is a PFC control method that activates a switching FET by detecting zero level of inductor current.

## 7.2 Configuration of 16-bit Timer KB20

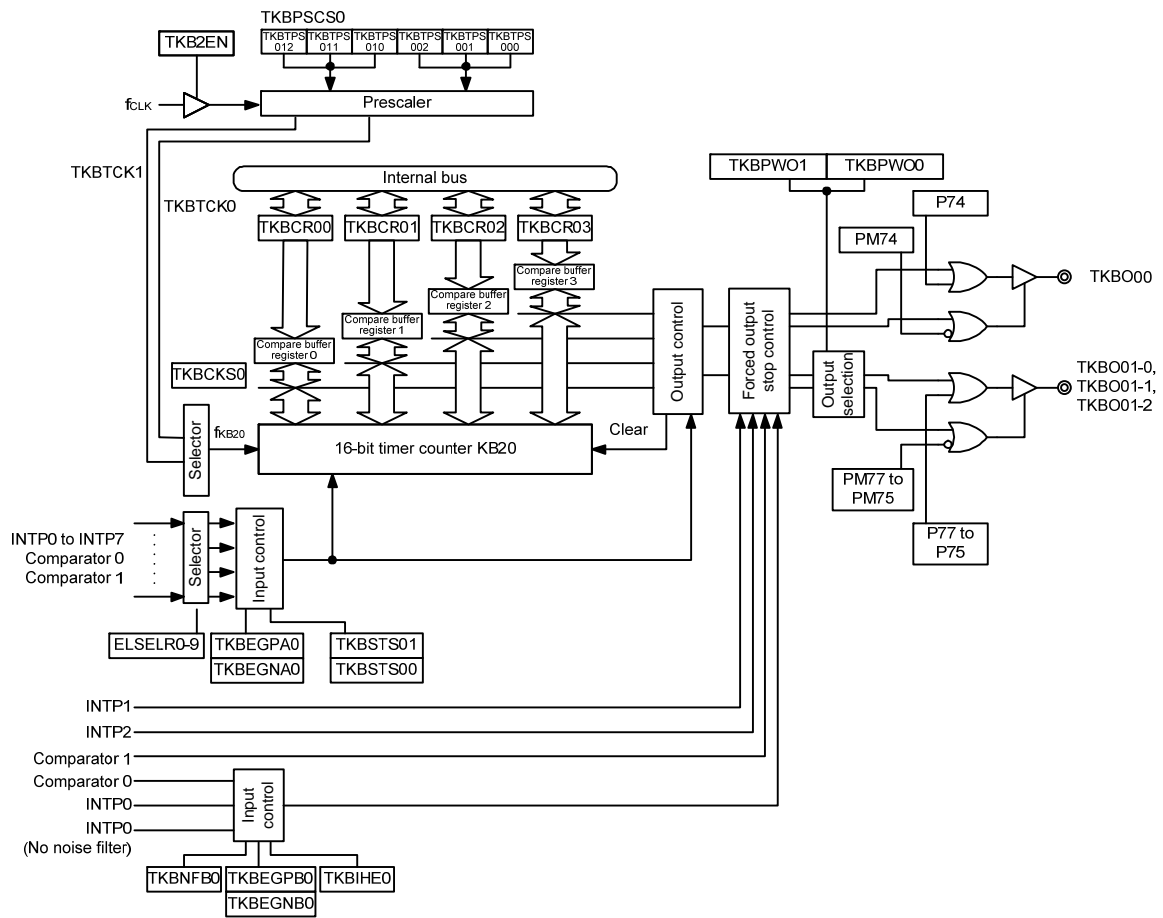
16-bit timer KB20 includes the following hardware.

**Table 7-1. Configuration of 16-bit Timer KB20**

Item	Configuration
Timer/counter	16-bit timer counter KB20 (TKBCNT0)
Registers	16-bit timer KB2 compare registers 00 to 03 (TKBCR00 to TKBCR03)
Timer output	TKBO00, TKBO01-0, TKBO01-1, TKBO01-2
Control registers	Peripheral enable register (PER1) Timer clock select register (TKBPSCS0) 16-bit timer KB2 operation control register 00 (TKBCTL00) 16-bit timer KB2 operation control register 01 (TKBCTL01) 16-bit timer KB2 output control register 00 (TKBIOC00) 16-bit timer KB2 output control register 01 (TKBIOC01) 16-bit timer KB2 flag register 0 (TKBFLG0) 16-bit timer KB2 trigger register 0 (TKBTRG0) 16-bit timer KB2 flag clear trigger register 0 (TKBCLR0) 16-bit timer KB2 dithering count registers 00, 01 (TKBDNR00, TKBDNR01) 16-bit timer KB2 compare 1L & dithering count register 00 (TKBCRLD00) 16-bit timer KB2 compare 3L & dithering count register 01 (TKBCRLD01) 16-bit timer KB2 smooth start initial duty registers 00, 01 (TKBSIR00, TKBSIR01) 16-bit timer KB2 smooth start step width registers 00, 01 (TKBSSR00, TKBSSR01) 16-bit timer KB2 maximum frequency limit setting register 0 (TKBMFR0) 16-bit timer KB2 counter restart select register (ELSELRn) 16-bit timer KB2 output switch register (PWCTKB) Forced output stop function control register 0 (TKBPACTL00) Forced output stop function control register 1 (TKBPACTL01) Forced output stop function control register 2 (TKBPACTL02) Forced output stop function flag register (TKBPAFLG0) Forced output stop function 1 start register (TKBPAHFS0) Forced output stop function 1 stop register (TKBPAHFT0) Port mode register 7 (PM7) Port register 7 (P7)

Figure 7-1 shows a block diagram.

Figure 7-1. Block Diagram of 16-bit Timer KB20



**Remark**  $f_{KB20}$ : Count clock of 16-bit timer KB20

**7.2.1 16-bit timer KB2 compare registers 00 to 03 (TKBCR00 to TKBCR03)**

TKBCR0m can be refreshed (writing the same value) and its value can be rewritten while the timer is counting (TKBCE0 = 1). When the value of TKBCR0m is rewritten while the timer is operating, that value is latched, transferred to TKBCR0m at the following timing, and the value of TKBCR0m is changed.

- When starting count operation of the counter (TKBCE0 = 0)
- When a batch overwrite trigger (TKBRDT0 = 1) or an external trigger (TKBTSE0 = 1) occurs

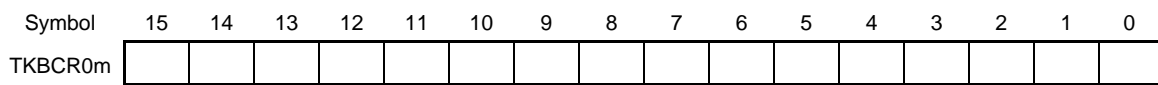
These registers can be set in 16-bit units.

Reset signal generation clears these registers to 0000H.

**Figure 7-2. Format of 16-bit Timer KB2 Compare Registers 00 to 03 (TKBCR00 to TKBCR03)**

Address: F0500H (TKBCR00), F0502H (TKBCR01), F0504H (TKBCR02), F0506H (TKBCR03)

After reset: 0000H R/W



**Remark** m = 0 to 3

### 7.3 Registers Controlling 16-bit Timer KB20

16-bit timer KB20 is controlled by the following registers.

- Peripheral enable register 1 (PER1)
- Timer clock select register (TKBPSCS0)
- 16-bit timer KB2 operation control register 00 (TKBCTL00)
- 16-bit timer KB2 operation control register 01 (TKBCTL01)
- 16-bit timer KB2 output control register 00 (TKBIOC00)
- 16-bit timer KB2 output control register 01 (TKBIOC01)
- 16-bit timer KB2 flag register 0 (TKBFLG0)
- 16-bit timer KB2 trigger register 0 (TKBTRG0)
- 16-bit timer KB2 flag clear trigger register 0 (TKBCLR0)
- 16-bit timer KB2 dithering count registers 00, 01 (TKBDNR00, TKBDNR01)
- 16-bit timer KB2 compare 1L & dithering count register 00 (TKBCRLD00)
- 16-bit timer KB2 compare 3L & dithering count register 01 (TKBCRLD01)
- 16-bit timer KB2 smooth start initial duty registers 00, 01 (TKBSIR00, TKBSIR01)
- 16-bit timer KB2 smooth start step width registers 00, 01 (TKBSSR00, TKBSSR01)
- 16-bit timer KB2 maximum frequency limit setting register 0 (TKBMFR0)
- 16-bit timer KB2 counter restart select register (ELSELRn)
- 16-bit timer KB2 output switch register (PWCTKB)
- Forced output stop function control register 0 (TKBPACTL00)
- Forced output stop function control register 1 (TKBPACTL01)
- Forced output stop function control register 2 (TKBPACTL02)
- Forced output stop function flag register (TKBPAFLG0)
- Forced output stop function 1 start register (TKBPAHFS0)
- Forced output stop function 1 stop register (TKBPAHFT0)
- Port mode register 7 (PM7)
- Port register 7 (P7)

### 7.3.1 Peripheral enable register (PER1)

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 timer KB20 is used, be sure to set bit 4 of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 7-3. Format of Peripheral Enable Register (PER1)**

Address: F0508H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
PER1	TMKAEN	0	CMPEN	TKB2EN	0	0	0	0

TKB2EN	Control of timer KB20 input clock
0	Stops supply of input clock. <ul style="list-style-type: none"> <li>• SFR used by timer KB20 cannot be written.</li> <li>• Timer KB20 is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by timer KB20 can be read/written.</li> </ul>

**Cautions 1.** When using 16-bit timer KB20, be sure to set the TKB2EN bit to 1 first and then set the following registers. If TKB2EN = 0, writing to a control register of 16-bit timer KB20 is ignored, and all read values are default values (except for 16-bit timer KB2 counter restart select register (ELSELRn), 16-bit timer KB2 output switch register (PWCTKB), and port mode register 7 (PM7)).

- Timer clock select register (TKBPSCS0)
- 16-bit timer KB2 operation control register 00 (TKBCTL00)
- 16-bit timer KB2 operation control register 01 (TKBCTL01)
- 16-bit timer KB2 output control register 00 (TKBIOC00)
- 16-bit timer KB2 output control register 01 (TKBIOC01)
- 16-bit timer KB2 flag register 0 (TKBFLG0)
- 16-bit timer KB2 trigger register 0 (TKBTRG0)
- 16-bit timer KB2 flag clear trigger register 0 (TKBCLR0)
- 16-bit timer KB2 dithering count registers 00, 01 (TKBDNR00, TKBDNR01)
- 16-bit timer KB2 compare 1L & dithering count register 00 (TKBCRLD00)
- 16-bit timer KB2 compare 3L & dithering count register 01 (TKBCRLD01)
- 16-bit timer KB2 smooth start initial duty registers 00, 01 (TKBSIR00, TKBSIR01)
- 16-bit timer KB2 smooth start step width registers 00, 01 (TKBSSR00, TKBSSR01)
- 16-bit timer KB2 maximum frequency limit setting register 0 (TKBMFR0)
- Forced output stop function control register 0 (TKBPACTL00)
- Forced output stop function control register 1 (TKBPACTL01)
- Forced output stop function control register 2 (TKBPACTL02)
- Forced output stop function flag register (TKBPAFLG0)
- Forced output stop function 1 start register (TKBPAHFS0)
- Forced output stop function 1 stop register (TKBPAHFT0)

2. Be sure to set bits 0 to 3 and 6 to "0".

### 7.3.2 Timer clock select register (TKBPSCS0)

The TKBPSCS0 register is a register that is used to select the division ratio of TKBTCK0/TKBTCK1.

Rewriting of the TKBPSCS0 register is possible only in the following cases.

If TKBTPS000 to TKBTPS002 can be rewritten:

All channels for which TKBTCK0 is selected as the operation clock (TKBCKS0 = 0) are stopped (TKBCE0 = 0).

If the TKBTPS010 to TKBTPS012 bits can be rewritten:

All channels for which TKBTCK1 is selected as the operation clock (TKBCKS0 = 1) are stopped (TKBCE0 = 0).

The TKBPSCS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 7-4. Format of Timer Clock Select Register (TKBPSCS0)**

Address: F052AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TKBPSCS0	0	TKBTPS012	TKBTPS011	TKBTPS010	0	TKBTPS002	TKBTPS001	TKBTPS000

TKBTPS0n2	TKBTPS0n1	TKBTPS0n0	Selection of operation clock (n = 0, 1) <sup>Note 1</sup>
0	0	0	f <sub>CLK</sub> or f <sub>HOCO</sub> with no division is selected for TKBTCKn <sup>Note 2</sup>
0	0	1	f <sub>CLK</sub> divided by 2 is selected for TKBTCKn <sup>Note 3</sup>
0	1	0	f <sub>CLK</sub> divided by 4 is selected for TKBTCKn <sup>Note 3</sup>
0	1	1	f <sub>CLK</sub> divided by 8 is selected for TKBTCKn <sup>Note 3</sup>
1	0	0	f <sub>CLK</sub> divided by 16 is selected as TKBTCKn <sup>Note 3</sup>
1	0	1	f <sub>CLK</sub> divided by 32 is selected as TKBTCKn <sup>Note 3</sup>
Other than above			Setting prohibited

- Notes**
- When changing the clock selected for f<sub>CLK</sub>, stop timer KB20 (TKBCE0 = 0).
  - f<sub>CLK</sub> is selected when FRQSEL4 = 0 and f<sub>HOCO</sub> is selected when FRQSEL4 = 1 in the user option byte (000C2H). When selecting f<sub>HOCO</sub> for the operating clock, set f<sub>CLK</sub> to f<sub>IH</sub> before setting bit 4 (TKB2EN) in peripheral enable register 1 (PER1). When changing f<sub>CLK</sub> to a clock other than f<sub>IH</sub>, clear bit 4 (TKB2EN) in peripheral enable register 1 (PER1) before changing.
  - Do not set this value when FRQSEL4 = 1 in the user option byte (000C2H).

**Caution** Be sure to clear bits 7 and 3 to 0.

**Remark** f<sub>CLK</sub>: CPU/peripheral hardware clock frequency



### 7.3.3 16-bit timer KB2 operation control register 00 (TKBCTL00)

TKBCTL00 is a register that is used to select the IH-only PWM output function, smooth start function, dithering function, maximum frequency limit function, interleaved PFC1 output, batch overwrite function for compare registers using external triggers, and counter triggers.

TKBCTL00 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

**Figure 7-5. Format of 16-bit Timer KB2 Operation Control Register 00 (TKBCTL00) (1/2)**

Address: F0522H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBCTL00	TKBIHE0	0	TKBSSE01	TKBDIE01	0	0	TKBSSE00	TKBDIE00
	7	6	5	4	3	2	1	0
	TKBMFE0	0	TKBIRS01	TKBIRS00	0	TKBTSE0	TKBSTS01	TKBSTS00
TKBIHE0	Control of IH-only PWM output function of TKBO01-0, TKBO01-1, TKBO01-2							
0	IH-only PWM output function not used							
1	IH-only PWM output function used							
TKBSSE0p	Control of PWM output smooth start function of TKBO00, TKBO01-0, TKBO01-1, TKBO01-2							
0	PWM output smooth start function not used							
1	PWM output smooth start function used							
TKBDIE0p	Control of PWM output dithering function of TKBO00, TKBO01-0, TKBO01-1, TKBO01-2							
0	PWM output dithering function not used							
1	PWM output dithering function used							
TKBMFE0	Control of maximum frequency limit function of TKBO00, TKBO01-0, TKBO01-1, TKBO01-2							
0	Maximum frequency limit function not used							
1	Maximum frequency limit function used							
TKBIRS01	TKBIRS00	Acceptable range setting of restart trigger source input for immediately outputting TKBO01-0, TKBO01-1, TKBO01-2 in interleave PFC output mode						
0	0	T/2 to T/2 + T/64						
0	1	T/2 to T/2 + T/32						
1	0	T/2 to T/2 + T/16						
1	1	T/2 to T/2 + T/8						
TKBTSE0	Control of compare register batch overwrite function set by external trigger							
0	Compare register batch overwrite function set by external trigger not used							
1	Compare register batch overwrite function set by external trigger used							

**Remark** p = 0, 1

**Figure 7-5. Format of 16-bit Timer KB2 Operation Control Register 00 (TKBCTL00) (2/2)**

Address: F0522H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBCTL00	TKBIHE0	0	TKBSSE01	TKBDIE01	0	0	TKBSSE00	TKBDIE00

	7	6	5	4	3	2	1	0
	TKBMFE0	0	TKBIRS01	TKBIRS00	0	TKBTSE0	TKBSTS01	TKBSTS00

TKBSTS01	TKBSTS00	Selection for timer KB20 restart trigger
0	0	Trigger input not used
0	1	Count restart trigger source 0 selected
1	0	Count restart trigger source 1 selected
1	1	Count restart trigger source 2 selected

- Cautions**
1. Do not rewrite the TKBCTL00 register during timer operation. However, the TKBCTL00 register can be refreshed (the same value can be written).
  2. Be sure to clear bits 14, 11, 10, 6, and 3 to 0.
  3. When using the PWM output function for IH control, set the TKBSTS01 and TKBSTS00 bits to 00B.

### 7.3.4 16-bit timer KB2 operation control register 01 (TKBCTL01)

TKBCTL01 is a register that controls the count operation and sets the count clock of the 16-bit timer.

TKBCTL01 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 7-6. Format of 16-bit Timer KB2 Operation Control Register 01 (TKBCTL01)**

Address: F0529H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
TKBCTL01	TKBCE0	0	0	TKBCKS00	0	0	TKBMD01	TKBMD00

TKBCE0	Control of timer KB20 operation
0	Timer operation stopped (counter is set to FFFF).
1	Timer count operation enabled

TKBCKS00	Selection of timer KB20 clock
0	TKBTCK0 selected
1	TKBTCK1 selected

TKBMD01	TKBMD00	Selection of timer KB20 operation mode
0	0	Standalone mode
1	1	Interleave PFC output mode
Other than above		Setting prohibited

- Cautions**
1. Do not rewrite the TKBCTL01 register during timer operation. However, the TKBCTL01 register can be refreshed (the same value can be written).
  2. In TKBCTL01, be sure to clear bits 6, 5, 3, and 2 to 0.

**7.3.5 16-bit timer KB2 output control register 00 (TKBIOC00)**

TKBIOC00 is a register that is used to set the default level/active level in 16-bit timer KB20 output (TKBO00, TKBO01-0, TKBO01-1, TKBO01-2).

TKBIOC00 can be set by a 1-bit or 8-bit memory manipulation instruction.  
Reset signal generation clears this register to 00H.

**Figure 7-7. Format of KB2 Output Control Register 00 (TKBIOC00)**

Address: F0526H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
TKBIOC00	0	0	0	0	TKBTOL01	TKBTOL00	TKBTOD01	TKBTOD00

TKBTOL0p	Active level setting of timer output TKBO00, TKBO01-0, TKBO01-1, TKBO01-2
0	Active level is set to high level
1	Active level is set to low level

TKBTOD0p	Default level setting of timer output TKBO00, TKBO01-0, TKBO01-1, TKBO01-2
0	Default level is set to low level
1	Default level is set to high level

- Cautions**
1. Do not rewrite the TKBIOC00 register during timer operation. However, the TKBIOC00 register can be refreshed (the same value can be written).
  2. Be sure to clear bits 7 to 4 to 0.
  3. Actual output of the TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 pins is set not only by TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output but by the port mode register (PM7) and port register (P7) for the shared ports.
  4. When using the PWM output function for IH control, set the TKBTOL01 and TKBTOD01 bits = 00B or TKBTOL01 and TKBTOD01 bits = 11B.

**Remark** p = 0, 1

**7.3.6 16-bit timer KB2 output control register 01 (TKBIOC01)**

TKBIOC01 is a register that controls output disabled/enabled of 16-bit timer KB20 output (TKBO00, TKBO01-0, TKBO01-1, TKBO01-2).

TKBIOC01 can be set by a 1-bit or 8-bit memory manipulation instruction.  
Reset signal generation clears this register to 00H.

**Figure 7-8. Format of 16-bit Timer KB2 Output Control Register 01 (TKBIOC01)**

Address: F0528H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TKBIOC01	TKBNFB0	0	TKBEGPA0	TKBEGNA0	TKBEGPB0	TKBEGNB0	TKBTOE01	TKBTOE00

TKBNFB0	Selection of forced output stop input 1 for PWM output function for IH control
0	For details on the settings, see 7.3.18 <b>Forced output stop function control register 0 (TKBPACTL00)</b> and 7.3.19 <b>Forced output stop function control register 1 (TKBPACTL01)</b> .
1	

TKBEGPA0	TKBEGNA0	Selection of active edge of restart trigger for PWM output function for IH control
0	0	No edge detected (restart trigger is invalid)
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

TKBEGPB0	TKBEGNB0	Selection of active edge of forced output stop input for PWM output function for IH control
0	0	No edge detected (forced output stop input is invalid)
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

TKBTOE0n	Output enabled/disabled of timer output TKBO00, TKBO01-0, TKBO01-1, TKBO01-2
0	Timer output disabled (Low-level output when TKBTOE0n = 0. High-level output when TKBTOE0n = 1.)
1	Timer output enabled

- Cautions**
1. Bits 1 and 0 (TKBTOE01, TKBTOE00) in the TKBIOC00 register can be rewritten during timer operation. However, the same value must be written to bits 7, and 5 to 2 when rewriting.
  2. Be sure to clear bit 6 to 0.
  3. Actual output of the TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 pins is set not only by TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output but by the port mode register (PM7) and the port register (P7) for the shared ports.

**Remark** n = 0, 1

**7.3.7 16-bit timer KB2 flag register 0 (TKBFLG0)**

TKBFLG0 is a register with status flags for 16-bit timer KB20.

TKBFLG0 can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 7-9. Format of 16-bit Timer KB2 Flag Register 0 (TKBFLG0)**

Address: F0513H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TKBFLG0	TKBSSF01	TKBSSF00	TKBSEF01	TKBSEF00	TKBIRF0	TKBIEF0	TKBMFF0	TKBRSF0

TKBSSF0n	Status flag for PWM output smooth start function of TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 pins
0	During stop in PWM output smooth start function
1	Executing in PWM output smooth start function

TKBSEF0n	Error flag for PWM output smooth start function of TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 pins
0	No error, or completion of clearing by TKBCLSEn
1	Error (TKBSSF0n = 1 occurred during PWM output smooth start execution (TKBRDT0 = 1))

TKBIRF0	Undetected restart trigger source 1 trigger error flag for interleave PFC mode
0	No error, or completion of clearing by TKBCLIR0
1	Error (Includes a period that restart trigger source 1 trigger is not detected in the range of $T/2 + T/n$ (n = 8, 16, 32, 64))

TKBIEF0	Restart trigger source 1 trigger multiplex detection error flag for interleave PFC mode
0	No error, or completion of clearing by TKBCLIE0
1	Error (Another count start trigger was detected during counting of the TKBO01-0, TKBO01-1, TKBO01-2 width)

TKBMFF0	Status flag for maximum frequency limit function
0	Maximum frequency limit function is not occurred, or completion of clearing by TKBCLMF0
1	Maximum frequency limit function is occurred

TKBRSF0	Batch overwrite trigger pending status flag
0	Batch overwrite enabled status or completion of batch overwrite caused by batch overwrite trigger
1	On hold (waiting for completion) status of batch overwrite due to writing to batch overwrite trigger bit TKBRDT0

- Remarks**
1. n = 0, 1
  2. T is the period of the last restart

### 7.3.8 16-bit timer KB2 trigger register 0 (TKBTRG0)

TKBTRG0 is a trigger register used for batch overwriting of the compare register for 16-bit timer KB20.

TKBTRG0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 7-10. Format of 16-bit Timer KB2 Trigger Register 0 (TKBTRG0)**

Address: F0512H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TKBTRG0	0	0	0	0	0	0	0	TKBRDT0

TKBRDT0	Trigger for batch overwrite request of compare register
0	Invalid setting
1	Batch overwrite request of compare register

**Remark** The read value of TKBTRG0 is always 0.

**7.3.9 16-bit timer KB2 flag clear trigger register 0 (TKBCLR0)**

TKBCLR0 is a register used to clear flags in 16-bit timer KB2 flag register 0 (TKBFLG0).

TKBCLR0 can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 7-11. Format of 16-bit Timer KB2 Flag Clear Trigger Register 0 (TKBCLR0)**

Address: F0527H After reset: 00H W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	0
TKBCLR0	0	0	TKBCLSE01	TKBCLSE00	TKBCLIR0	TKBCLIE0	TKBCLMF0	0

TKBCLSE0n	Trigger for clearing error flag for PWM output smooth start function of TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 pins
0	Invalid setting
1	Clear the TKBSEF01 flag to 0.

TKBCLIR0	Trigger for clearing undetected restart trigger source 1 trigger error flag for interleave PFC mode
0	Invalid setting
1	Clear the TKBIRF0 flag to 0.

TKBCLIE0	Trigger for clearing restart trigger source 1 trigger multiplex detection error flag for interleave PFC mode
0	Invalid setting
1	Clear the TKBIEF0 flag to 0.

TKBCLMF0	Trigger for clearing status flag for maximum frequency limit function
0	Invalid setting
1	Clear the TKBMFF0 flag to 0.

**Caution** Be sure to clear bits 7 and 6 to 0.

- Remarks**
1. n = 0, 1
  2. The read value of TKBTRG0 is always 0.



**7.3.10 16-bit timer KB2 dithering count registers 00, 01 (TKBDNR00, TKBDNR01)**

TKBDNR0p is a register that is used by the PWM dithering function for TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output.

When the values of the higher 4 bits of this register are N (N = 0H to FH), the active period of N period cycles during the 16-period cycle of PWM output is output by extending one clock.

Figure 7-13 shows the relation among the TKBDNR0p setting and the repetitions (N) of the period cycle extending the active period by one clock and the ordinal of the period (the kth period) during the 16-period cycle to be extended.

TKBDNR0p can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 7-12. Format of 16-bit Timer KB2 Dithering Count Registers 00, 01 (TKBDNR00, TKBDNR01)**

Address: F050EH (TKBDNR00), F0510H (TKBDNR01) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TKBDNR0p					0	0	0	0

**Caution** Be sure to clear bits 3 to 0 to 0. The TKBDNR0p register can be rewritten during timer operation.

**Remark** p = 0, 1

**Figure 7-13. 16-bit Timer KB2 Dithering Count Register 0p (TKBDNR0p) Setting**

kth period Repetitions (N)	k															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0																
1	■															
2	■							■								
3	■				■			■								
4	■				■			■				■				
5	■		■		■			■				■				
6	■		■		■			■			■		■			
7	■		■		■		■				■		■			
8	■		■		■		■				■		■		■	
9	■	■			■		■				■		■		■	
10	■	■			■		■			■		■		■		■
11	■	■			■		■			■		■		■		■
12	■	■	■				■			■		■		■		■
13	■	■	■		■		■			■		■		■		■
14	■	■	■	■			■			■		■		■		■
15	■	■	■	■	■					■		■		■		■

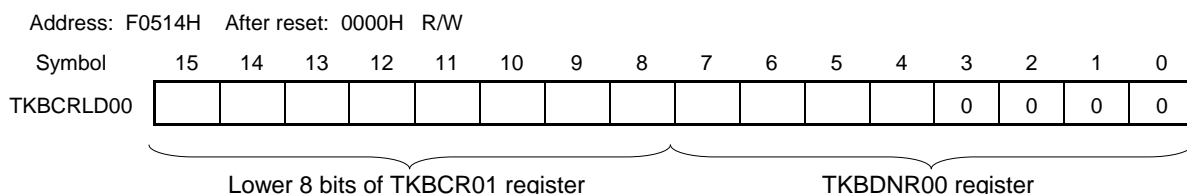
- Remarks 1.**  cell: Set to active period according to settings in TKBCR01 and TKBCR03 registers  
 cell: Set to active period according to "settings + 1" in TKB0CR1 and TKBCR03 registers
- 2.** p = 0, 1

**7.3.11 16-bit timer KB2 compare 1L & dithering count register 00 (TKBCRLD00)**

TKBCRLD00 is a register that stores the “lower 8 bits of TKBCR01 register” values in its higher 8 bits and the “TKBDNR00 register” values in its lower 8 bits.

TKBCRLD00 can be set by a 16-bit memory manipulation instruction.  
 Reset signal generation clears this register to 0000H.

**Figure 7-14. Format of 16-bit Timer KB2 Compare 1L & Dithering Count Register 00 (TKBCRLD00)**



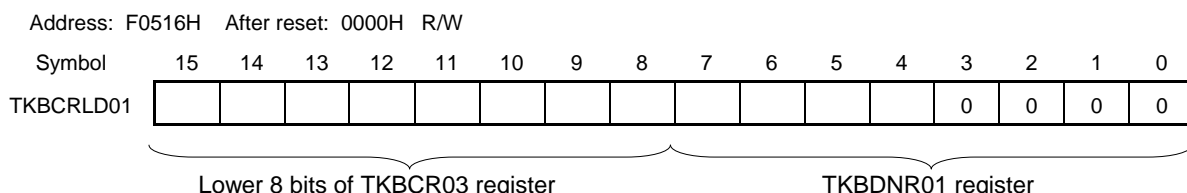
**Caution** Be sure to clear bits 3 to 0 to 0. The TKBDNR0p register can be rewritten during timer operation.

**7.3.12 16-bit timer KB2 compare 3L & dithering count register 01 (TKBCRLD01)**

TKBCRLD01 is a register that stores the “lower 8 bits of TKBCR03 register” values in its higher 8 bits and the “TKBDNR01 register” values in its lower 8 bits.

TKBCRLD01 can be set by a 16-bit memory manipulation instruction.  
 Reset signal generation clears this register to 0000H.

**Figure 7-15. Format of 16-bit Timer KB2 Compare 3L & Dithering Count Register 01 (TKBCRLD01)**



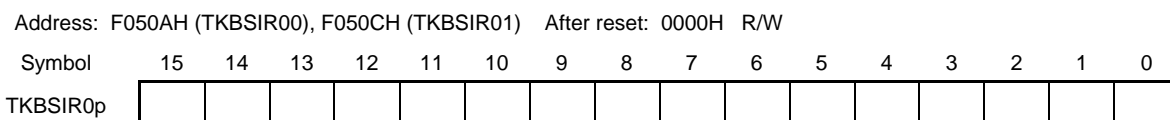
**Caution** Be sure to clear bits 3 to 0 to 0. The TKBDNR0p register can be rewritten during timer operation.

**7.3.13 16-bit timer KB2 smooth start initial duty registers 00, 01 (TKBSIR00, TKBSIR01)**

TKBSIR0p is a register that sets the default duty for the PWM output smooth start function for TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output.

TKBSIR0p can be set by a 16-bit memory manipulation instruction.  
Reset signal generation clears this register to 0000H.

**Figure 7-16. Format of 16-bit Timer KB2 Smooth Start Initial Duty Registers 00, 01 (TKBSIR00, TKBSIR01)**



**Caution** The TKBSIR0p can be rewritten during timer operation.

**Remark** p = 0, 1

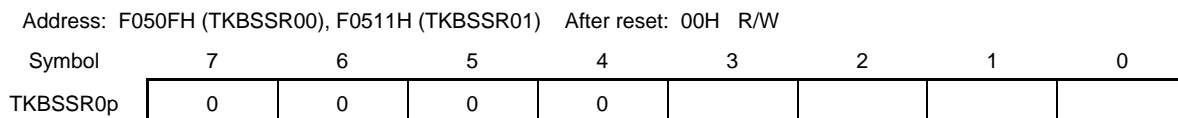
**7.3.14 16-bit timer KB2 smooth start step width registers 00, 01 (TKBSSR00, TKBSSR01)**

TKBSSR0p is a register that is used by the PWM output smooth start function for TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output.

When the value of this register is N (N = 0000B to 1111B), output of a PWM with the active output period is continued for N + 1 times by setting TKBSIR0p. Afterward, output continues with the (active period + 1 clock) waveform for N + 1 cycles, then with the (active period + 2 clock) waveform for N + 1 cycles, and so on. Finally, when TKBCR01 and TKBCR03 have the same duty, the PWM output smooth start function is cleared and normal PWM output is set.

TKBSSR0p can be set by an 8-bit memory manipulation instruction.  
Reset signal generation clears this register to 00H.

**Figure 7-17. Format of 16-bit Timer KB2 Smooth Start Step Width Registers 00, 01 (TKBSSR00, TKBSSR01)**



**Caution** Be sure to clear bits 7 to 4 to 0. The TKBSSR0p can be rewritten during timer operation.

**Remark** p = 0, 1

### 7.3.15 16-bit timer KB2 maximum frequency limit setting register 0 (TKBMFR0)

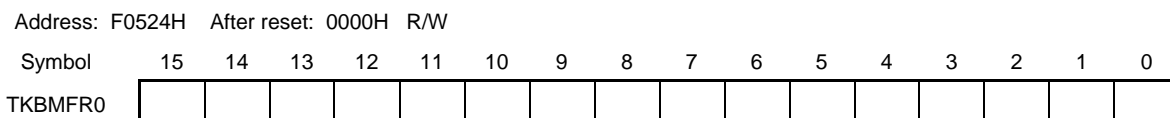
TKBMFR0 is a register that sets the minimum period for the timer restart of external trigger.

When the counter (TKBCNT0) value is smaller than this TKBMFR0 value, if trigger input is detected, the trigger is held pending, and the counter (TKBCNT0) is cleared (restart) after counting to the value set to TKBMFR0.

TKBMFR0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

**Figure 7-18. Format of 16-bit Timer KB2 Maximum Frequency Limit Setting Register 0 (TKBMFR0)**



Do not rewrite the TKBMFR0 register during timer operation. However, the TKBMFR0 register can be refreshed (the same value can be written).

**7.3.16 16-bit timer KB2 counter restart select register (ELSELRn)**

ELSELRn is a register that is used to assign counter restart trigger sources 2 to 0 and the PWM output function for IH control restart trigger source to each external restart trigger (external triggers 7 to 0 and comparator detection 1, 0).

Reset signal generation clears this register to 00H.

**Figure 7-19. Format of 16-bit Timer KB2 Counter Restart Select Register (ELSELRn)**

Address: F01E0H (ELSELR00), F01E1H (ELSELR01), F01E2H (ELSELR02), F01E3H (ELSELR03)  
 F01E4H (ELSELR04), F01E5H (ELSELR05), F01E6H (ELSELR06), F01E7H (ELSELR07)  
 F01E8H (ELSELR08), F01E9H (ELSELR09)                      After reset: 00H                      R/W

Symbol	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	0	ELSELRn2	ELSELRn1	ELSELRn0

ELSELRn2	ELSELRn1	ELSELRn0	Restart Source	Operation after Restart
0	0	0	None	None
0	0	1	TMKB2 counter restart trigger source 0	Timer output restart
0	1	0	TMKB2 counter restart trigger source 1	Batch overwrite of compare registers
0	1	1	TMKB2 counter restart trigger source 2	
1	0	0	TMBK2 IH-PWM output restart request signal	IH-PWM output restart
Other than above			Setting prohibited	

**Caution** Be sure to set bits 7 to 3 to 0. Do not rewrite the ELSELRn register during timer operation. However, the ELSELRn register can be refreshed (the same value can be written).

**Remark** n = 00 to 09

The following lists the correspondence between each event and the ELSELRn register.

Count Restart Trigger Input	Corresponding Register
External interrupt 0 (INTP0)	ELSELR00
External interrupt 1 (INTP1)	ELSELR01
External interrupt 2 (INTP2)	ELSELR02
External interrupt 3 (INTP3)	ELSELR03
External interrupt 4 (INTP4)	ELSELR04
External interrupt 5 (INTP5)	ELSELR05
External interrupt 6 (INTP6)	ELSELR06
External interrupt 7 (INTP7)	ELSELR07
Comparator detection 0	ELSELR08
Comparator detection 1	ELSELR09

- Cautions**
1. Do not specify multiple count restart trigger inputs as one restart trigger source.
  2. When the signal output from comparator 0 or 1 is selected as the IH-PWM output restart request signal, set the C1EDG, C1EPO, C0EDG, and C0EPO bits of the comparator filter control register (COMPFIR) to "0".
  3. The external interrupt request signals (INTP0 to INTP7), which are used as TMKB2 count restart triggers 0 to 2 and IH-PWM output restart request signals, are not affected by the setting of the external interrupt rising edge enable register (EGP0) and external interrupt falling edge enable register (EGN0).

<R>

**7.3.17 16-bit timer KB2 output switch register (PWCTKB)**

This register is used to select the output of the TKBO01-0, TKBO01-1, and TKBO01-2 pins.

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

Reset signal generation clears this register to 00H.

**Figure 7-20. Format of 16-bit Timer KB2 Output Switch Register (PWCTKB)**

Address: F007CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PWCTKB	0	0	0	0	0	0	TKBPWO1	TKBPWO0

TKBPWO1	TKBPWO0	Timer KB2 output switch
0	0	TKBO01-0 selected for output
0	1	TKBO01-1 selected for output
1	0	TKBO01-2 selected for output
1	1	Setting prohibited

- Cautions**
1. Do not rewrite the PWCTKB register during timer operation. However, the PWCTKB register can be refreshed (the same value can be written).
  2. Be sure to set bits 7 to 2 to 0.

**7.3.18 Forced output stop function control register 0 (TKBPACTL00)**

TKBPACTL00 is a register that selects the signal to be used as the trigger to control the forced output stop function for the TKBO00 pin.

TKBPACTL00 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

**Figure 7-21. Format of Forced Output Stop Function Control Register 0 (TKBPACTL00) (1/2)**

Address: F0530H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBPACTL00	TKBPAFXS003	TKBPAFXS002	TKBPAFXS001	TKBPAFXS000	0	0	0	TKBPAFCM00
	7	6	5	4	3	2	1	0
	0	TKBPAHVS002	TKBPAHVS001	TKBPAHVS000	TKBPAHCM00	TKBPAHCM00	TKBPAMD001	TKBPAMD000
					1	0		
TKBPAFXS003		Trigger input selection for forced output stop function 2 (3)						
0		INTP2 not used as trigger						
1		INTP2 used as trigger						
TKBPAFXS002		Trigger input selection for forced output stop function 2 (2)						
0		INTP1 not used as trigger						
1		INTP1 used as trigger						
TKBPAFXS001		Trigger input selection for forced output stop function 2 (1)						
0		Comparator detection 1 not used as trigger						
1		Comparator detection 1 used as trigger						
TKBPAFXS000	TKBCTL00	Trigger input for forced output stop function 2 (0)						
	TKBIHE0							
0		INTP0 and comparator 0 detection not used as trigger						
1		Comparator detection 0 used as trigger						
Other than above		Setting prohibited						
TKBPAFCM00		Operation mode selection for forced output stop function 2						
0		Forced output stop function 2 is started when forced output stop input 2 is detected. Forced output stop function 2 is cancelled in synchronization with the next restart of the counter.						
1		Forced output stop function 2 is started when forced output stop input 2 is detected. After cancellation of the trigger is detected, forced output stop function is cancelled in synchronization with the next restart of the counter.						
TKBPAHVS002		Trigger input selection for forced output stop function 1 (2)						
0		INTP1 not used as trigger						
1		INTP1 used as trigger						

**Caution** The TKBPACTL00 register cannot be used when using the PWM output function for IH control. When using the PWM output function for IH control, be sure to set this register to its initial value (0000H).

**Figure 7-21. Format of Forced Output Stop Function Control Register 0 (TKBPACTL00) (2/2)**

Address: F0530H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBPACTL00	TKBPAFXS003	TKBPAFXS002	TKBPAFXS001	TKBPAFXS000	0	0	0	TKBPAFCM00
	7	6	5	4	3	2	1	0
	0	TKBPAHZS002	TKBPAHZS001	TKBPAHZS000	TKBPAHCM00	TKBPAHCM00	TKBPAMD001	TKBPAMD000
					1	0		
TKBPAHZS000	TKBCTL00	Trigger input selection for forced output stop function 1 (0)						
	TKBIHE0							
0	–	INTP0 and comparator 0 detection not used as trigger						
1	0	Comparator detection 0 used as trigger						
Other than above		Setting prohibited						
TKBPAHCM00	TKBPAHCM00	Operation mode selection for forced output stop function 1						
1	0							
0	0	Forced output stop function 1 is started when forced output stop input 1 is detected. Forced output stop function 1 is cancelled when TKBPAHTT0 is set to 1, regardless of the level of the input.						
0	1	Forced output stop function 1 is started when forced output stop input 1 is detected. After the input is cancelled, forced output stop function 1 is cancelled when TKBPAHTT0 is set to 1. Setting TKBPAHTT0 = 1 is invalid during the active period of the input.						
1	0	Forced output stop function 1 is started when forced output stop input 1 is detected. After TKBPAHTT0 is set to 1, forced output stop function 1 is cancelled in synchronization with the next restart of the counter, regardless of the level of the input.						
1	1	Forced output stop function 1 is started when forced output stop input 1 is detected. After the input is cancelled, forced output stop function 1 is cancelled in synchronization with the next restart of the counter after TKBPAHTT0 is set to 1. Setting TKBPAHTT0 = 1 is invalid during the active period of the input.						
TKBPAMD001	TKBPAMD000	Output status selection when executing forced output stop function						
		Forced output stop function 1			Forced output stop function 2			
0	0	High-impedance output			Output fixed at low level			
0	1	High-impedance output			Output fixed at high level			
1	0	Output fixed at low level			Output fixed at low level			
1	1	Output fixed at high level			Output fixed at high level			

- Cautions**
1. The TKBPACTL00 register cannot be used when using the PWM output function for IH control. When using the PWM output function for IH control, be sure to set this register to its initial value (0000H).
  2. Be sure to clear bits 11 to 9 and 7 to 0. Do not rewrite the TKBPACTL00 register during timer operation. However, the TKBPACTL00 register can be refreshed (the same value can be written).
  3. INTP1 and INTP2, which are used to trigger forced output stop functions 1 and 2, are not affected by the setting of the external interrupt rising edge enable register (EGP0) and external interrupt falling edge enable register (EGN0). Only the rising edge is valid. Use the TKBEGPB0 and TKBEGNB0 bits of 16-bit timer KB2 output control register 01 (TKBIOC01) to select the valid edge of the INTP0 signal used by forced output stop functions 1 and 2.



**7.3.19 Forced output stop function control register 1 (TKBPACTL01)**

TKBPACTL01 is a register that selects the signal to be used as the trigger to control the forced output stop function for the TKBO01-0, TKBO01-1, and TKBO01-2 pins.

TKBPACTL01 can be set by a 16-bit memory manipulation instruction.  
Reset signal generation clears this register to 0000H.

**Figure 7-22. Format of Forced Output Stop Function Control Register 1 (TKBPACTL01) (1/2)**

Address: F0532H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBPACTL01	TKBPAFXS013	TKBPAFXS012	TKBPAFXS011	TKBPAFXS010	0	0	0	TKBPAFCM01
	7	6	5	4	3	2	1	0
	0	TKBPAHXS012	TKBPAHXS011	TKBPAHXS010	TKBPAHCM01	TKBPAHCM01	TKBPAMD011	TKBPAMD010
					1	0		

TKBPAFXS013	Trigger input selection for forced output stop function 2 (3)
0	INTP2 not used as trigger
1	INTP2 used as trigger

TKBPAFXS012	Trigger input selection for forced output stop function 2 (2)
0	INTP1 not used as trigger
1	INTP1 used as trigger

TKBPAFXS011	Trigger input selection for forced output stop function 2 (1)
0	Comparator detection 1 not used as trigger
1	Comparator detection 1 used as trigger

TKBPAFXS010	TKBCTL00	Trigger input for forced output stop function 2 (0)
	TKBIHE0	
0	–	INTP0 and comparator 0 detection not used as trigger
1	0	Comparator detection 0 used as trigger
Other than above		Setting prohibited

TKBPAFCM01	Operation mode selection for forced output stop function 2
0	Forced output stop function 2 is started when forced output stop input 2 is detected. Forced output stop function 2 is cancelled in synchronization with the next restart of the counter.
1	Forced output stop function 2 is started when forced output stop input 2 is detected. After the reverse edge of the trigger is detected, forced output stop function 2 is cancelled in synchronization with the next restart of the counter.

TKBPAHXS012	Input selection for forced output stop function 1 (2)
0	INTP1 not used as trigger
1	INTP1 used as trigger

**Caution** Only bit 4 (TKBPAHXS010) of the TKBPACTL01 register can be used when using the PWM output function for IH control. When using the PWM output function for IH control, be sure to set the other bits in this register to their initial value (0).

**Figure 7-22. Format of Forced Output Stop Function Control Register 1 (TKBPACTL01) (2/2)**

Address: F0532H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBPACTL01	TKBPAFXS013	TKBPAFXS012	TKBPAFXS011	TKBPAFXS010	0	0	0	TKBPAFCM01
	7	6	5	4	3	2	1	0
	0	TKBPAHZS012	TKBPAHZS011	TKBPAHZS010	TKBPAHCM01	TKBPAHCM01	TKBPAMD011	TKBPAMD010
					1	0		

TKBPAHZS011	Trigger input selection for forced output stop function 1 (1) <sup>Note</sup>
0	Comparator detection 1 not used as trigger
1	Comparator detection 1 used as trigger

TKBPAHZS010	TKBCTL00	TKBIOC01	Trigger input selection for forced output stop function 1 (0) <sup>Note</sup>
	TKBIHE0	TKBNFB0	
0	–	–	INTP0 and comparator 0 detection not used as trigger
1	0	–	Comparator detection 0 used as trigger
1	1	0	INTP0 without noise filtering used as trigger
1	1	1	INTP0 with noise filtered used as trigger

TKBPAHCM01	TKBPAHCM01	Operation mode selection for forced output stop function 1
1	0	
0	0	Forced output stop function 1 is started when forced output stop input 1 is detected. Forced output stop function 1 is cancelled when TKBPAHTT01 is set to 1, regardless of the level of the input.
0	1	Forced output stop function 1 is started when forced output stop input 1 is detected. After the input is cancelled, forced output stop function 1 is cancelled when TKBPAHTT01 is set to 1. Setting TKBPAHTT01 = 1 invalid during the active period of the input.
1	0	Forced output stop function 1 is started when forced output stop input 1 is detected. After TKBPAHTT1 is set to 1, forced output stop function 1 is cancelled with the next restart of the counter, regardless of the level of the input.
1	1	Forced output stop function 1 is started when forced output stop input 1 is detected. After the input is cancelled, forced output stop function 1 is cancelled with the next restart of the counter after TKBPAHTT1 is set to 1. Setting TKBPAHTT1 = 1 is invalid during the active period of the input.

TKBPAMD011	TKBPAMD010	Output status selection when executing forced output stop function	
		Forced output stop function 1	Forced output stop function 2
0	0	High-impedance output	Output fixed at low level
0	1	High-impedance output	Output fixed at high level
1	0	Output fixed at low level	Output fixed at low level
1	1	Output fixed at high level	Output fixed at high level

**Note** When using INTP0 as a trigger, input it for at least two f<sub>CLK</sub> cycles. In addition, setting TKBPAHCM010 to 1 is prohibited when using the comparator 0 or 1 detection signal as a trigger while the C0EDG and C1EDG bits of the comparator filter control register (COMPFIR) are 1 (both-edge detection). Be sure to set TKBPAHCM010 to 0.

- Cautions**
1. Be sure to clear bits 11 to 9 and 7 to 0. Do not rewrite the TKBPACTL00 register during timer operation. However, the TKBPACTL00 register can be refreshed (the same value can be written).
  2. INTP1 and INTP2, which are used to trigger forced output stop functions 1 and 2, are not affected by the setting of the external interrupt rising edge enable register (EGP0) and external interrupt falling edge enable register (EGN0). Only the rising edge is valid. Use the TKBEGPB0 and TKBEGNB0 bits of 16-bit timer KB2 output control register 01 (TKBIOC01) to select the valid edge of the INTP0 signal used by forced output stop function 1.
  3. Only bit 4 (TKBPAHVS010) of the TKBPACTL01 register can be used when using the PWM output function for IH control. When using the PWM output function for IH control, be sure to set the other bits in this register to their initial value (0).

### 7.3.20 Forced output stop function control register 2 (TKBPACTL02)

TKBPACTL02 is a register that enables or disables the forced output stop function.

TKBPACTL02 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 7-23. Format of Forced Output Stop Function Control Register 2 (TKBPACTL02)**

Address: F0537H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TKBPACTL02	0	0	0	0	0	0	TKBPACE01	TKBPACE00

TKBPACE01	Control of forced output stop function for TKBO01-0, TKBO01-1, TKBO01-2 pins
0	Disables the forced output stop function.
1	Enables the forced output stop function.

TKBPACE00	Control of forced output stop function for TKBO00 pin
0	Disables the forced output stop function.
1	Enables the forced output stop function.

**Cautions** 1. Be sure to clear bits 7 to 2 to 0. The TKBPACTL02 register can be rewritten during timer operation.

2. Bit 0 (TKBPAHTT00) of the TKBPAHFT0 register cannot be used when using the PWM output function for IH control. When using the PWM output function for IH control, be sure to set this bit to its initial value (0).

**7.3.21 Forced output stop function flag register (TKBPAFLG0)**

TKBPAFLG0 is a register with status flags for the forced output stop function.

TKBPAFLG0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 7-24. Format of Forced Output Stop Function Flag Register (TKBPAFLG0)**

Address: F0536H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TKBPAFLG0	TKBPAFSF01	TKBPAHSF01	TKBPAFSF00	TKBPAHSF00	TKBPAFIF01	TKBPAHIF01	TKBPAFIF00	TKBPAHIF00

TKBPAFSF01	Status flag of forced output stop function 2 for TKBO01-0, TKBO01-1, TKBO01-2 pins
0	Forced output stop canceled
1	Forced output stopped

TKBPAHSF01	Status flag of forced output stop function 1 for TKBO01-0, TKBO01-1, TKBO01-2 pins
0	Forced output stop canceled
1	Forced output stop stopped

TKBPAFSF00	Status flag of forced output stop function 2 for TKBO00 pin
0	Forced output stop canceled
1	Forced output stop stopped

TKBPAHSF00	Status flag of forced output stop function 1 for TKBO00 pin
0	Forced output stop cancel status
1	Forced output stop status

TKBPAFIF01	Status flag of forced output stop input 2 for TKBO01-0, TKBO01-1, TKBO01-2 pins
0	Forced output stop input 2 is inactive level
1	Forced output stop input 2 is active level

TKBPAHIF01	Status flag of forced output stop input 1 for TKBO01-0, TKBO01-1, TKBO01-2 pins
0	Forced output stop input 1 is inactive level
1	Forced output stop input 1 is active level

TKBPAFIF00	Status flag of forced output stop input 2 for TKBO00 pin
0	Forced output stop input 2 is inactive level
1	Forced output stop input 2 is active level

TKBPAHIF00	Status flag of forced output stop input 1 for TKBO00 pin
0	Forced output stop input 1 is inactive level
1	Forced output stop input 1 is active level

- Cautions**
1. Status flags TKBPAHIF00, TKBPAFIF00, TKBPAHIF01, and TKBPAFIF01 cannot be used when using the comparator 0 or 1 detection signal as a trigger while the C0EDG and C1EDG bits of the comparator filter control register (COMPFIR) are 1 (both-edge detection).
  2. Only bit 6 (TKBPAHSF01) of the TKBPAFLG0 register can be used when using the PWM output function for IH control. Do not use the other status flags.

**7.3.22 Forced output stop function 1 start register (TKBPAHFS0)**

TKBPAHFS0 is used to start forced output stop function 1.  
 TKBPAHFS0 can be set by a 1-bit or 8-bit memory manipulation instruction.  
 Reset signal generation clears this register to 00H.

**Figure 7-25. Format of Forced Output Stop Function 1 Start Register (TKBPAHFS0)**

Address: F0534H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TKBPAHFS0	0	0	0	0	0	0	TKBPAHTS01	TKBPAHTS00

TKBPAHTS01	Start of forced output stop function 1 for TKBO01-0, TKBO01-1, TKBO01-2 pins
0	Writing 0 is invalid.
1	Starts forced output stop function 1 for TKBO01-0, TKBO01-1, TKBO01-2 pins.

TKBPAHTS00	Start of forced output stop function 1 for TKBO00 pin
0	Writing 0 is invalid.
1	Starts forced output stop function 1 for TKBO00 pin.

- Cautions**
1. Be sure to clear bits 7 to 2 to 0. The TKBPAHFS0 register can be rewritten during timer operation.  
 The read value is 0.
  2. The TKBPAHFS0 register cannot be used when using the PWM output function for IH control. When using the PWM output function for IH control, be sure to set this register to its initial value (00H).

**Remark** The read value of TKBPAHFS0 is always 0.

**7.3.23 Forced output stop function 1 stop register (TKBPAHFT0)**

TKBPAHFT0 is used to stop forced output stop function 1.  
 TKBPAHFT0 can be set by a 1-bit or 8-bit memory manipulation instruction.  
 Reset signal generation clears this register to 00H.

**Figure 7-26. Format of Forced Output Stop Function 1 Stop Register (TKBPAHFT0)**

Address: F0535H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TKBPAHFT0	0	0	0	0	0	0	TKBPAHTT01	TKBPAHTT00

TKBPAHTT01	Stop of forced output stop function 1 for TKBO01-0, TKBO01-1, TKBO01-2 pins
0	Writing 0 is invalid.
1	Stops forced output stop function 1 for TKBO01-0, TKBO01-1, TKBO01-2 pins.

TKBPAHTT00	Stop of forced output stop function 1 for TKBO00 pin
0	Writing 0 is invalid.
1	Stops forced output stop function 1 for TKBO00 pin.

- Cautions**
1. Be sure to clear bits 7 to 2 to 0. The TKBPAHFT0 register can be rewritten during timer operation.  
 The read value is 0.
  2. Bit 0 (TKBPAHTT00) of the TKBPAHFT0 register cannot be used when using the PWM output function for IH control. When using the PWM output function for IH control, be sure to set this bit to its initial value (0).

**Remark** The read value of TKBPAHFT0 is always 0.

When the TKBPAHCM001 and TKBPAHCM000 bits are 10 or 11, forced output function 1 is cancelled when the TMKB hardware macro period is generated after TKBPAHTT0n is set to 1.

For details on the operation when forced output stop input is detected or TKBPAHTS0n is set to 1 during the period after TKBPAHTT0n is set to 1 and before the TMKB hardware macro period is generated, see **7.7.3 Notes on using forced output stop function 1.**

#### 7.3.24 Registers controlling port functions of 16-bit timer KB2 output pins

- <R> When using 16-bit timer KB20, set the registers that control the port functions multiplexed on the target pins (LCD port function register (PFSEGx), port mode register (PMxx), and port register (Pxx)). For details, see **4.3.9 LCD port function register (PFSEG0 to PFSEG6)**, **4.3.1 Port mode registers (PMxx)**, and **4.3.2 Port registers (Pxx)**.
- <R> When using the ports (such as P74/TKBO00) to be shared with the 16-bit timer KB20 output pins for 16-bit timer KB20 output, set the LCD port function register (PFSEGx) bit, port mode register (PMxx) bit, and port register (Pxx) bit corresponding to each port to 0.
- <R> Example: When using P74/TKBO00 for 16-bit timer KB20 output
- Set the PFSEG16 bit of LCD port function register (PFSEG2) to 0.
  - Set the PM74 bit of port mode register (PM7) to 0.
  - Set the P74 bit of port register (P7) to 0.



## 7.4 Operation of 16-bit Timer KB20

Operation specifications of 16-bit timer KB20 are described below.

- Counter basic operation (see 7.4.1)
- Default level and active level (see 7.4.2)
- Stop/restart operation (see 7.4.3)
- Batch overwrite operation (see 7.4.4)

There are three operation modes for 16-bit timer KB20.

- Standalone mode (period controlled by TKBCR00) (see 7.4.5)
- Standalone mode (period controlled by external trigger input) (see 7.4.6)
- Interleave PFC (power factor correction) output mode (see 7.4.7)

### 7.4.1 Counter basic operation

#### (1) Count start operation

In any mode, the 16-bit counter of timer KB20 starts counting from its initial value (FFFFH). The counter increments from FFFFH to 0000H, 0001H, 0002H, 0003H and so on.

#### (2) Clear operation

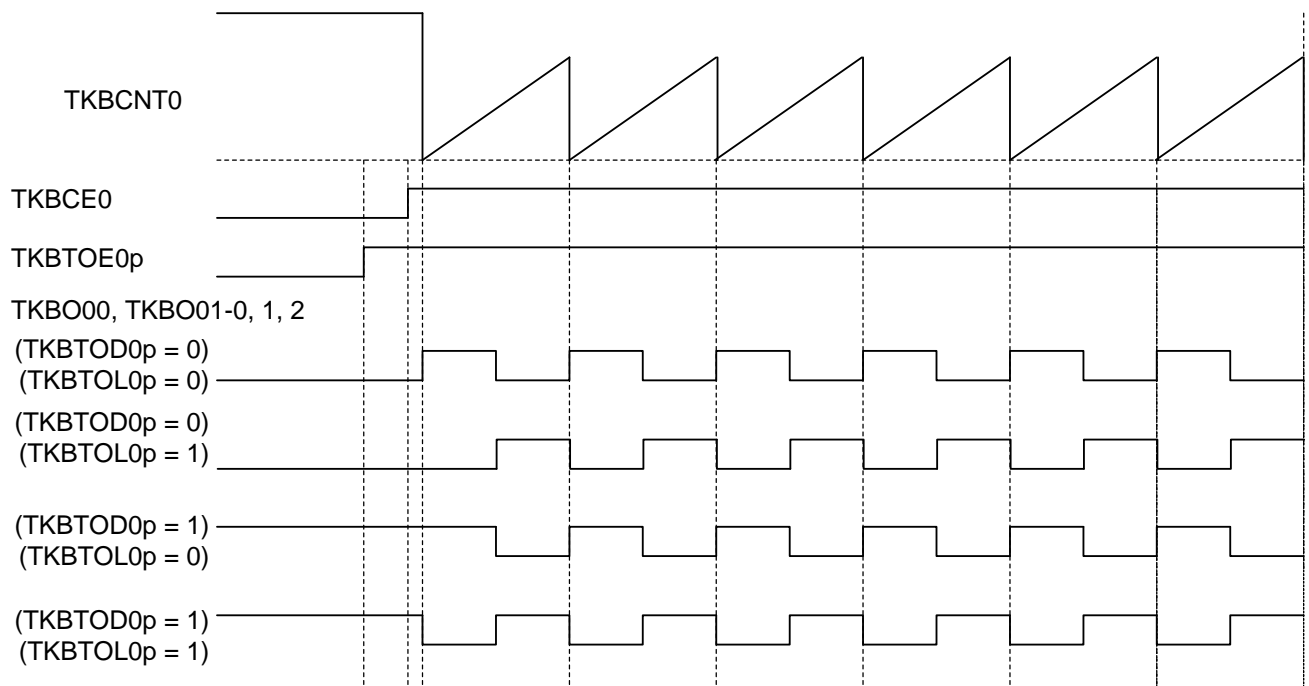
The 16-bit counter is cleared to 0000H when the 16-bit counter value matches the value defined in TKBCR00 or an external trigger is in effect if the period is determined by external triggers. An INTTKB2 interrupt occurs when the counter is cleared at the time when it matches the value defined in TKBCR00, but it does not occur when the counter is cleared by an external trigger.

7.4.2 Default level and active level

(1) Basic operation

The default level and active level can be specified for timer KB20 output by using 16-bit timer KB2 output control register 00 (TKBIOC00).

Figure 7-27. Timing of Default Level and Active Level (Basic Operation)



When TKBTOE0p is changed from 0 to 1, output from the TKBO00, TKBO01-0, TKBO01-1, and TKBO01-2 pins is enabled and the PWM waveform is output according to the TKBTOL0p setting.  
 When TKBTOE0p is changed from 1 to 0, output from TKBO01-0, TKBO01-1, and TKBO01-2 pins is disabled and the default level is output according to the TKBTOD0p setting.

**Remark** p = 0, 1

**Caution** When using the PWM output function for IH control, set the TKBTOL01 and TKBTOD01 bits = 00B or TKBTOL01 and TKBTOD01 bits = 11B.

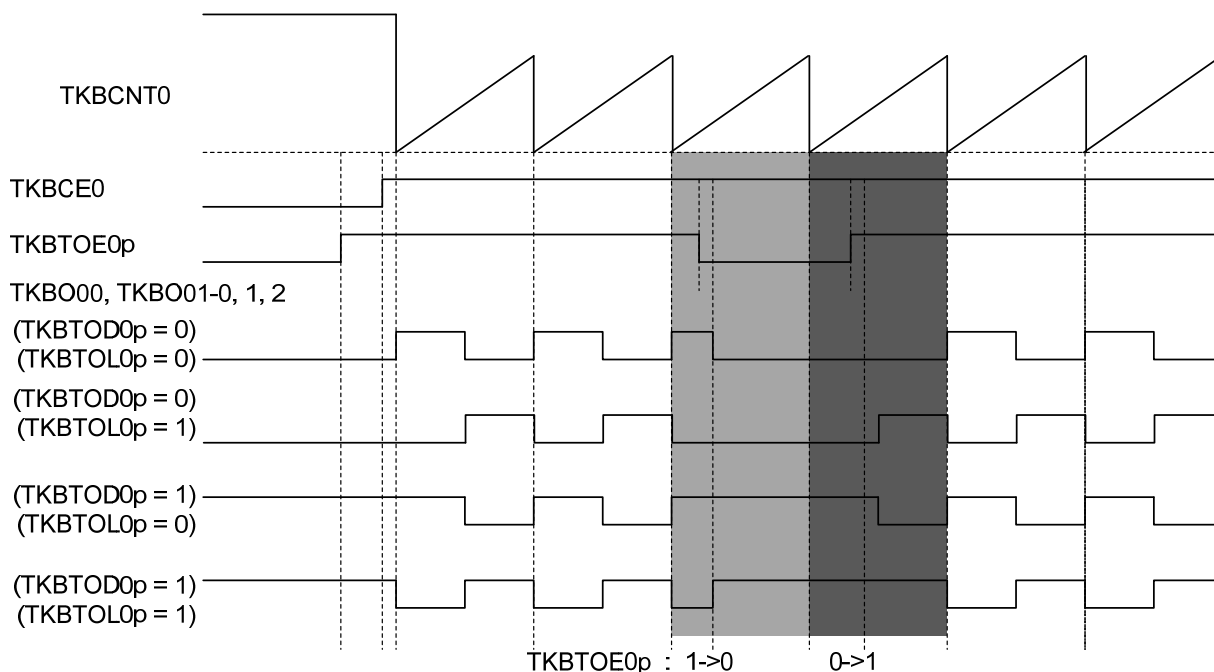
**(2) When TKBTOE0p is changed from 0 to 1**

When TKBTOE0p is changed from 0 to 1 before the value of counter TKBCNT0 matches the value of the compare register (TKBCR01 to TKBCR03) while the timer counter is operating, the timer outputs the PWM waveform in accordance with the TKBTOL0p setting when the values match.

If TKBTOE0p is changed from 0 to 1 after the value of counter TKBCNT0 matches the value of compare register (TKBCR01 to TKBCR03), the timer output remains its default level until the next restart of the counter.

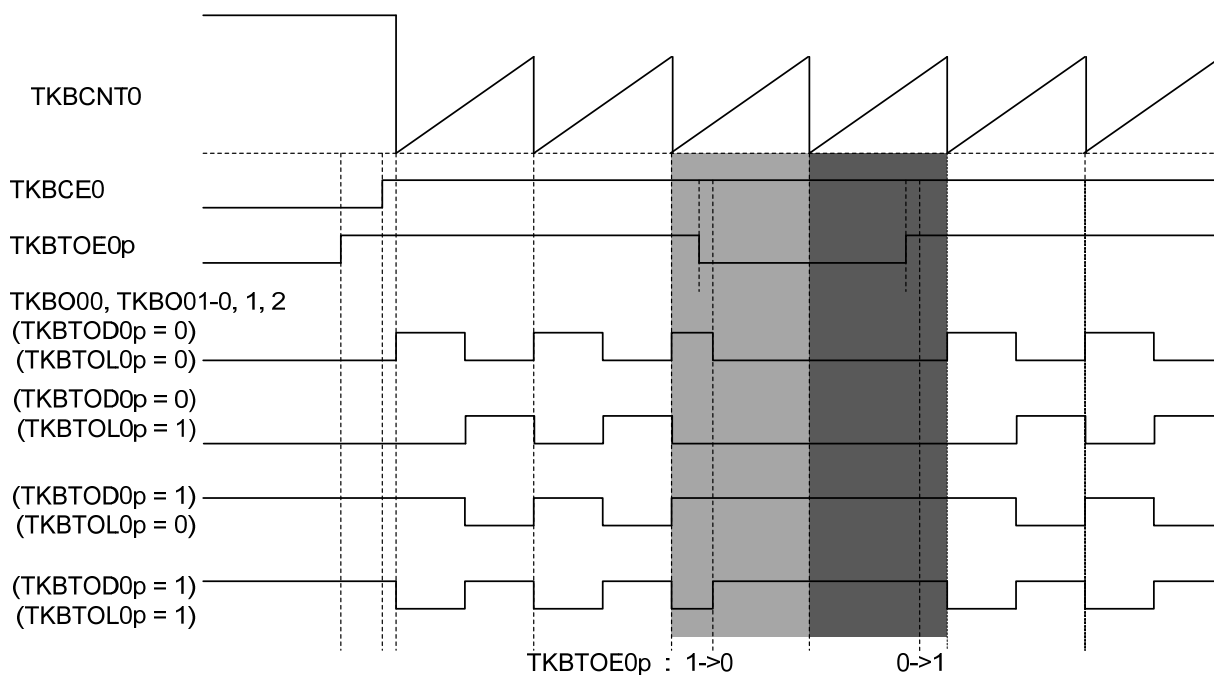
**Figure 7-28. Timing of Default Level and Active Level**

**(When TKBTOE0p is changed from 0 to 1 before counter and compare register (TKBCR01 to TKBCR03) values match)**



**Figure 7-29. Timing of Default Level and Active Level**

**(When TKBTOE0p is changed from 0 to 1 after counter and compare register (TKBCR01 to TKBCR03) values match)**



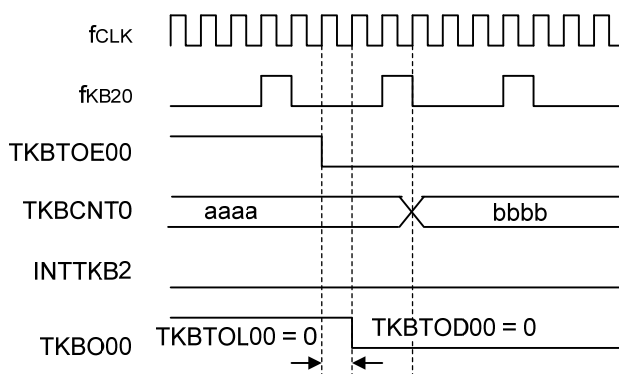
**Remark** p = 0, 1

**(3) When TKBTOE0p is changed from 1 to 0**

**(a) Basic timing**

TKBO00, TKBO01-0, TKBO01-1, and TKBO01-2 output the default level set by TKBTOD0p 1 fCLK cycle after KBTOE0p is changed from 1 to 0.

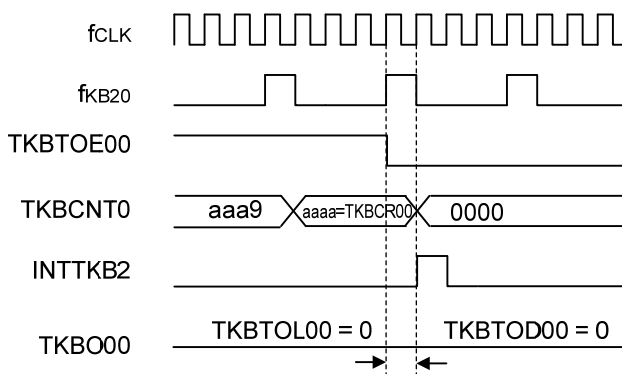
**Figure 7-30. Timing of Default Level and Active Level  
(When TKBTOE00 is changed from 1 to 0)**



**(b) When TKBCNT0 and TKBCR00 match at the same time as TKBTOE0p is cleared:**

When TKBTOE0p is changed from 1 to 0 at the same time as TKBCNT0 and TKBCR00 match, the change of TKBTOE0p is given priority, and TKBO00, TKBO01-0, TKBO01-1, and TKBO01-2 keep outputting the default level set by TKBTOD0p.

**Figure 7-31. Timing of Default Level and Active Level  
(When TKBTOE0p is changed from 1 to 0 at the same time as TKBCNT0 and TKBCR00 match)**



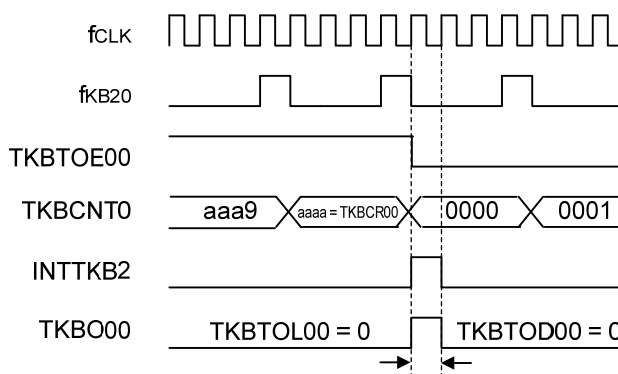
**Remark** p = 0, 1

**(c) When TKBTOE0p is manipulated at the same time as the timer count clock is generated**

When TKBTOE0p is manipulated at the same time as f<sub>KB20</sub> is generated, TKBO00, TKBO01-0, TKBO01-1, and TKBO01-2 are set when TKBCNT0 and TKBCR00 match.

One f<sub>CLK</sub> cycle later, TKBO00, TKBO01-0, TKBO01-1, and TKBO01-2 start outputting the default level set by TKBTOD0p.

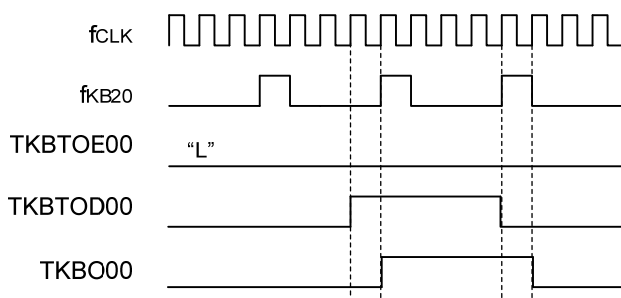
**Figure 7-32. Timing of Default Level and Active Level  
(When TKBTOE00 is manipulated at the same time as the timer count clock is generated)**



**(4) When TKBTOD0p is changed while TKBTOE0p is 0**

When TKBTOD0p is changed while TKBTOE0p is 0, TKBO00, TKBO01-0, TKBO01-1, and TKBO01-2 output the default level set by TKBTOD0p 1 f<sub>CLK</sub> cycle later.

**Figure 7-33. Timing of Default Level and Active Level  
(When TKBTOD00 is changed while TKBTOE00 is 0)**



**Remark** p = 0, 1

**7.4.3 Stop/restart operation**

16-bit timer KB20 can be stopped and started by controlling TKBCE0.

16-bit timer KB20 is reset and stops operating when TKBCE0 is changed from 1 to 0.

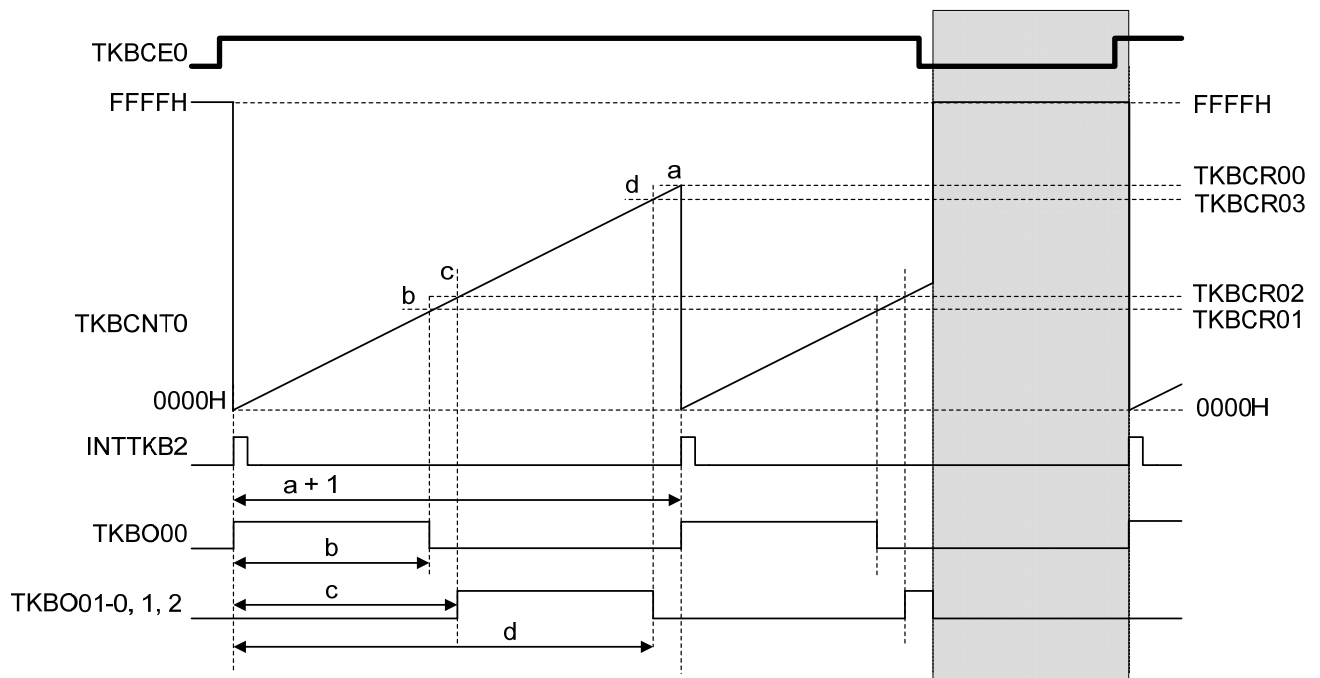
Counter TKBCNT0 is reset to FFFFH and then stops operating.

TKBO00, TKBO01-0, TKBO01-1, and TKBO01-2 output the default level set by TKBTOD0p.

16-bit timer KB20 starts operating when TKBCE0 is changed from 0 to 1.

Counter TKBCNT0 maintains FFFFH while TKBCE0 is 0 and starts up counting when TKBCE0 is changed from 0 to 1.

**Figure 7-34. Timing of Stop Operation (When TKBTOL0p = 0 and TKBTOD0p = 0)**

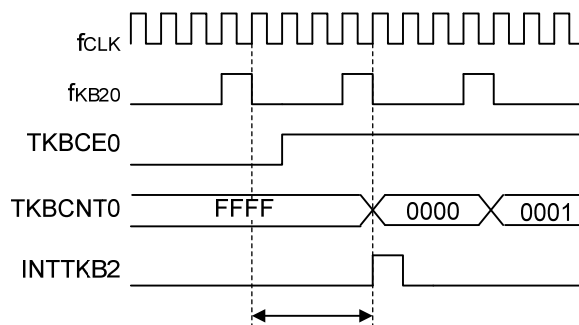


**Remark** p = 0, 1

**(1) Counting start timing**

Counting starts one  $f_{CLK}$  cycle (min.) to one  $f_{KB20}$  cycle (max.) after  $TKBCE0$  is changed from 0 to 1.  $INTTMKB2$  is output when counting starts.

**Figure 7-35. Timing of Start Operation (When  $TKBCE0$  is changed from 0 to 1)**

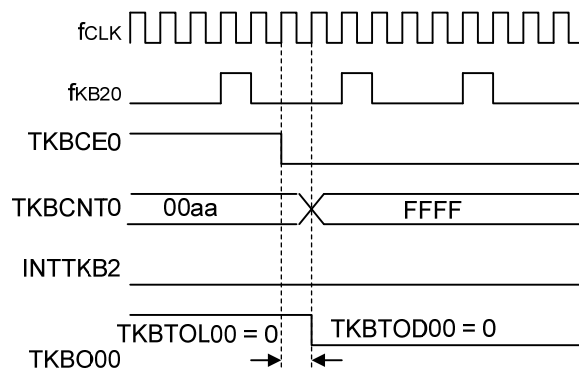


**(2) Counting stop timing**

**(a) Basic timing**

Counting stops one  $f_{CLK}$  cycle after  $TKBCE0$  is changed from 1 to 0.  $TKBCNT0$  is reset to FFFFH and  $TKBO00$ ,  $TKBO01-0$ ,  $TKBO01-1$ , and  $TKBO01-2$  output the default level set by  $TKBTOD0p$ .

**Figure 7-36. Timing of Stop Operation (When  $TKBCE0$  is changed from 1 to 0)**



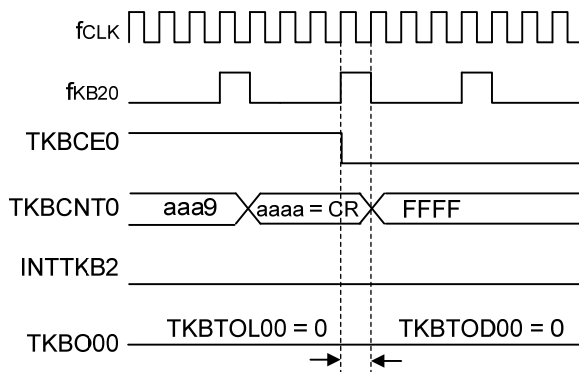
**Remark**  $p = 0, 1$

**(b) When TKBCNT0 and TKBCR00 match at the same time as TKBTOE0p is cleared**

When TKBCE0 is changed from 1 to 0 at the same time as TKBCNT0 and TKBCR00 match, the change of TKBCNT0 is given priority, and TKBO00, TKBO01-0, TKBO01-1, and TKBO01-2 keep outputting their default level set by TKBTOD0p. At this time, INTTKB2 is not generated.

**Figure 7-37. Timing of Stop Operation**

**(When TKBCE0 is changed from 1 to 0 at the same time as TKBCNT0 and TKBCR00 match)**



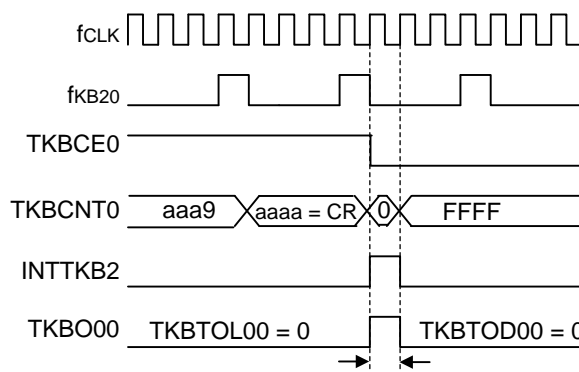
**(c) When TKBCE0 is manipulated at the same time as the timer count clock is generated**

When TKBCE0 is manipulated at the same time as fCLK is generated, INTTKB2 is output when TKBCNT0 and TKBCR00 match, and then TKBO00, TKBO01-0, TKBO01-1, and TKBO01-2 are set.

One fCLK cycle later, TKBCNT0 is reset to FFFFH and TKBO00, TKBO01-0, TKBO01-1, and TKBO01-2 output the default level set by TKBTOD0p.

**Figure 7-38. Timing of Stop Operation**

**(When TKBCE0 is manipulated at the same time as the timer count clock is generated)**



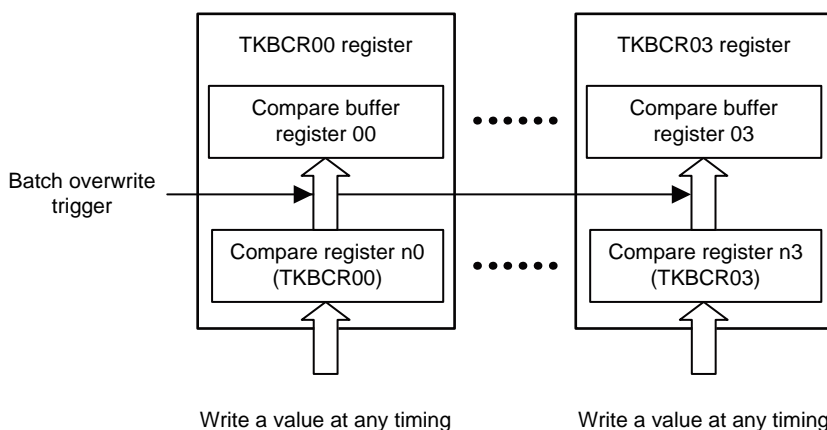
**Remark** p = 0, 1



**7.4.4 Batch overwrite operation**

As shown in Figure 7-39, 16-bit timer KB2 compare register 0p (TKBCR0p) for 16-bit timer KB20 has two stages. Therefore, its value is not applied immediately even if any value is set to TKBCR0p by a program. The value set to TKBCR0p at any timing is transferred at once to buffer registers when the counter starts running or when a transfer trigger occurs, and it is actually used for any comparison operation. This enables multiple compare registers to be set with each value at any timing.

**Figure 7-39. Compare Register Batch Overwrite Function**



**Remark** As shown above, 16-bit timer KB2 compare register 0p (TKBCR0p) has two stages and is treated as a single register except when values are written to.

**(1) Timing of batch overwrite**

There are three cases when the compare registers are written all together. Among these, (c) can be controlled by configuration of the register.

- (a) When starting count operation of 16-bit timer KB20
- (b) The count value of the 16-bit counter and the value that is set to 16-bit timer KB2 compare register 00 (TKBCR00) match.
- (c) An external trigger occurs while batch overwrite with an external trigger is enabled.

**Remark** p = 0 to 3

7.4.5 Standalone mode (period controlled by TKBCR00)

(1) Outline of functions

In standalone operation mode, the period is defined according to the value of TKBCR00, TKBO00 is generated by TKBCR00 and TKBCR01, and then TKBO01-0, TKBO01-1, and TKBO01-2 are generated by TKBCR02 and TKBCR03.

The duty can be set within a range of 0% to 100% and the period and duty can be calculated using the following formula.

[Calculation Formula for TKBO00 Output]

Pulse period = (TKBCR00 setting + 1) × Count clock period

Duty [%] = (TKBCR01 setting / (TKBCR00 setting + 1)) × 100

0% output: TKBCR01 setting = 0000H

100% output: TKBCR01 setting ≥ TKBCR00 setting + 1

[Calculation Formula for TKBO01-0, TKBO01-1, TKBO01-2 Output]

Duty [%] = ((TKBCR03 setting – TKBCR02 setting) / (TKBCR00 setting + 1)) × 100

0% output: TKBCR03 setting = TKBCR02 setting

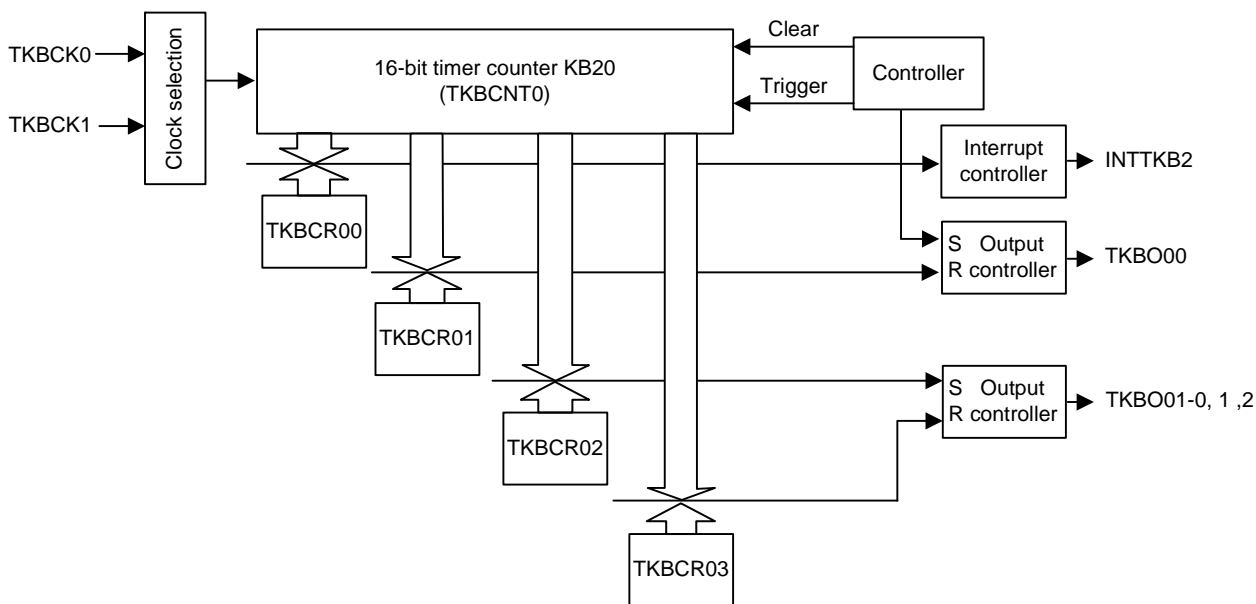
100% output: TKBCR02 setting = 0000H

TKBCR03 setting ≥ TKBCR00 setting + 1

**Caution** It should always be: TKBCR02 setting ≤ TKBCR03 setting.

Figure 7-40 shows the configuration of standalone mode (period controlled by TKBCR00).

Figure 7-40. Configuration of Standalone Mode (Period Controlled by TKBCR00)

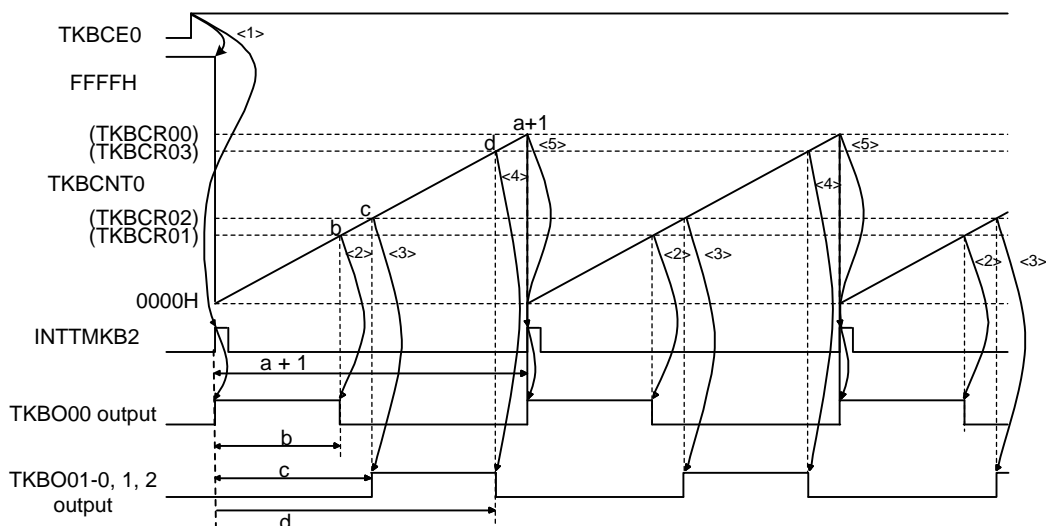


**Remark** Output from TKBO01-0, TKBO01-1, and TKBO01-2 is switched by the 16-bit timer KB2 switch register (PWCTKB).

**(2) Outline of operation**

Figure 7-41 shows the timing sample for standalone mode.

**Figure 7-41. Timing Sample for Standalone Mode (Period Controlled by TKBCR00)  
(When default value of output is low level (TKBTOD0p = 0) and active level is high level (TKBTOL0p = 0))**



This section describes an example about the standalone operation (periodic controlled by TKBCR00). The following descriptions are linked with <1> to <5> in Figure 7-41.

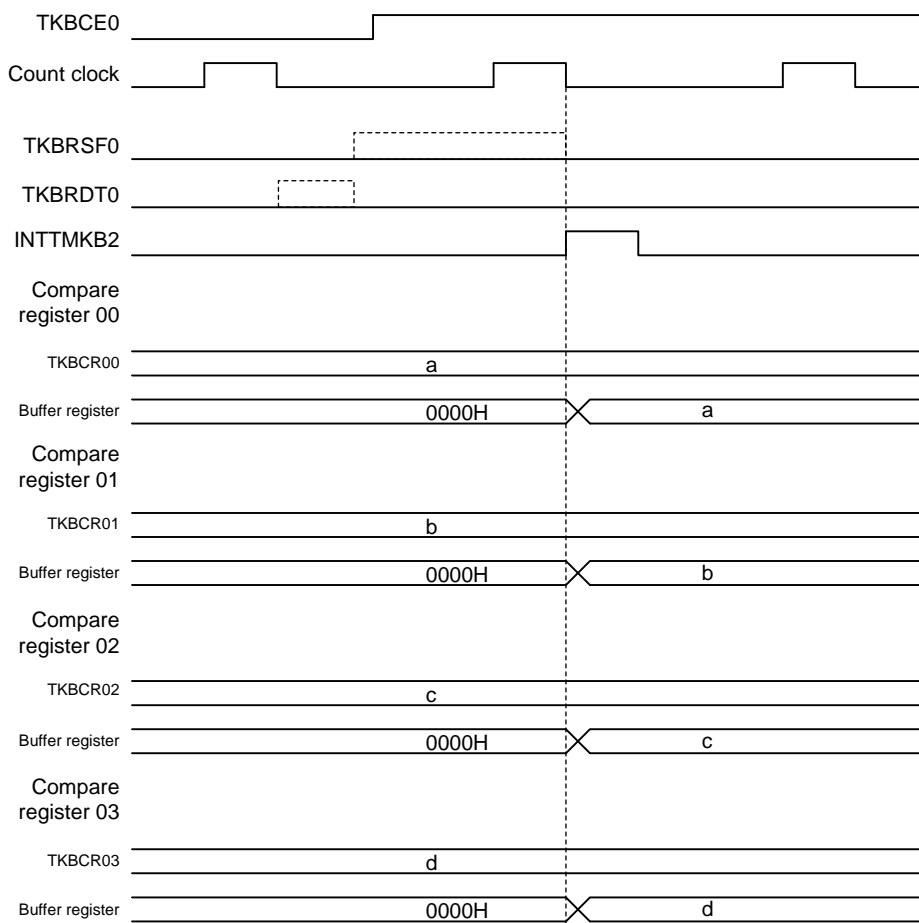
- <1> When TKBCE0 is set to 1, 16-bit timer counter KB2 (TKBCNT0) changes from FFFFH to 0000H in synchronization with the count clock, then it starts counting up. At the same time, INTTMKB2 output is generated and TKBO00 output changes from its default value specified with the TKBTOD00 bit in TKB0IOC0 register to its active value (high level in this example) specified with the TKBTOL00 bit (TKBO01-0, TKBO01-1, TKBO01-2 output holds its default value specified with the TKBTOD01 bit).
- <2> When TKBCNT0 is counted up and its value matches the value specified in 16-bit timer KB2 compare register 01 (TKBCR01), TKBO00 output becomes inactive level.
- <3> When TKBCNT0 is counted up and its value matches the value specified in 16-bit timer KB2 compare register 02 (TKBCR02), TKBO01-0, TKBO01-1, TKBO01-2 output becomes active level.
- <4> When TKBCNT0 is counted up and its value matches the value specified in 16-bit timer KB2 compare register 03 (TKBCR03), TKBO01-0, TKBO01-1, TKBO01-2 output becomes inactive level.
- <5> When TKBCNT0 is counted up and its value matches the value specified in 16-bit timer KB2 compare register 00 (TKBCR00), INTTMKB2 output is generated at the next count clock and TKBO00 output becomes active level. TKBCNT0 starts its upward counting from 0000H.
- <6> Repeats <2> through <5>.

**(3) Operation of batch overwrite (at starting counting operation)**

The compare register of 16-bit timer KB20 has a function which updates the internal buffer registers simultaneously at the starting of counter operation caused by the count clock which is generated after writing 1 to the TKBCE0 bit in the TKBCTL01 register.

Batch overwrite is generated even 1 is not written to the TKBRD0 bit in the TKBTRG0 register only in the case of counting operation start timing (see Figure 7-42).

**Figure 7-42. Batch Overwrite Function: Buffer Updating Timing at Counting Operation Start**



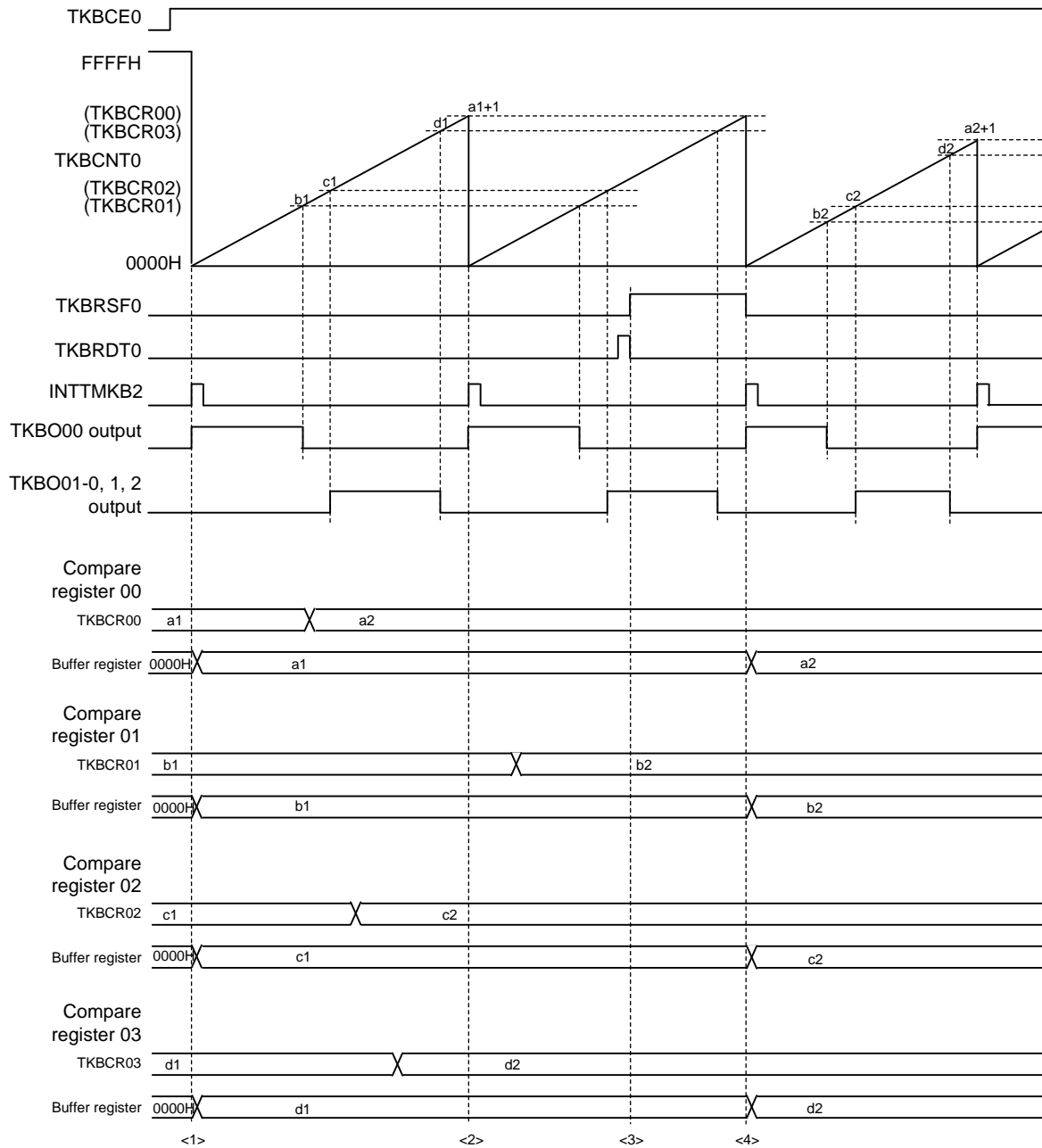
**Remark** TKBRSF0 is set to 1 by writing 1 to TKBRD0 when TKBCE0 is 0. TKBRSF0 is cleared to 0 when counting starts (counter start trigger generated).

**(4) Batch overwrite operation (buffer updating during counting operation)**

The compare register of 16-bit timer KB20 has a function which updates the internal buffer register simultaneously at the next counter clear (TKBCNT0 and TKBCR00 matched), identifying writing 1 to the TKBRDT0 bit as batch overwriting trigger. The batch overwrite trigger pending status flag (TKBRSF0) is set during the period from writing 1 to the TKBRDT0 bit until completion of batch overwrite (see Figure 7-43).

- <1> Compare register setting is transferred to the buffer register at the timing when the TKBCE0 bit is set from 0 to 1 and TKBCNT0 starts counting operation.
- <2> After the TKBCR00 to TKBCR03 registers are overwritten, even when counter clear is generated, batch overwrite is not generated if 1 is not written to the TKBRDT0 bit.
- <3> The batch overwrite trigger pending status flag (TKBRSF0 bit) is set to 1 by writing 1 to the TKBRDT0 bit.
- <4> Compare register setting is transferred to the buffer register by the counter clear generated when the TKBRSF0 bit is 1. The TKBRSF0 bit is set to 0 simultaneously.

Figure 7-43. Batch Overwrite Function: Timing of Buffer Updating during Counting Operation



(5) Sample of register setting details at standalone mode (period controlled by TKBCR00)

	15	14	13	12	11	10	9	8
TKBCTL00	TKBIHE0 1/0	– 0	TKBSSE01 1/0	TKBDIE01 1/0	– 0	– 0	TKBSSE00 1/0	TKBDIE00 1/0
	7	6	5	4	3	2	1	0
	TKBMFE0 0	– 0	TKBIRS01 0	TKBIRS00 0	– 0	TKBTSE0 0	TKBSTS01 0	TKBSTS00 0
TKBCTL01	TKBCE0 1	– 0	– 0	TKBCKS0 1/0	– 0	– 0	TKBMD01 0	TKBMD00 0
	7	6	5	4	3	2	1	0
TKBIOC00	– 0	– 0	– 0	– 0	TKBTOL01 1/0	TKBTOL00 1/0	TKBTOD01 1/0	TKBTOD00 1/0
	7	6	5	4	3	2	1	0
TKBIOC01	TKBNFB0 0	– 0	TKBEGPA0 1/0	TKBEGNA0 1/0	TKBEGPB0 1/0	TKBEGNB0 1/0	TKBTOE01 1/0	TKBTOE00 1/0
	7	6	5	4	3	2	1	0
TKBPSCS0	– 0	TKBTPS012 1/0	TKBTPS011 1/0	TKBTPS010 1/0	– 0	TKBTPS002 1/0	TKBTPS001 1/0	TKBTPS000 1/0
TKBCR00	0000H to FFFFH							
TKBCR01	0000H to FFFFH							
TKBCR02	0000H to FFFFH							
TKBCR03	0000H to FFFFH							
TKBSIR00	0000H to FFFFH							
TKBSIR01	0000H to FFFFH							
TKBSSR00	00H to 0FH							
TKBSSR01	00H to 0FH							
TKBDNR00	00H to F0H							
TKBDNR01	00H to F0H							
TKBMFR0	0000H							

: Setting is fixed for this mode       : Setting is not needed (default setting)

### 7.4.6 Standalone mode (period controlled by external trigger input)

#### (1) Outline of functions

In standalone mode, the period can be controlled not only by TKBCR00 but also by external trigger input.

The input signals selected by 16-bit timer KB2 counter restart trigger select registers 00 to 09 (ELSELR00 to ELSELR09) and the TKBSTS01 and TKBSTS00 bits in 16-bit timer KB2 operation control register 00 (TKBCTL00) are used to detect external trigger input.

When the external trigger input is detected, counter TKBCNT0 is cleared to 0000H and TKBO00/TKBO01-0, TKBO01-1, TKBO01-2 output is respectively set to active level and inactive level. When the setting values of TKBCR00 and the counter (TKBCNT0) match before detection of external trigger input, the counter is cleared to 0000H and operation is continued.

For the formula to calculate TKBO00/TKBO01-0, TKBO01-1, TKBO01-2 output when external trigger input is not yet detected and the period is controlled by TKBCR00, see **7.4.5 Standalone mode (period controlled by TKBCR00)**.

The calculation formula for TKBO00/TKBO01-0, TKBO01-1, TKBO01-2 output when the period is controlled by external trigger input detection is as follows:

[Calculation Formula for TKBO00 Output]

Pulse period = (Counter value at external trigger input detection + 1) × Count clock period

Duty [%] = (Setting value of TKBCR01 / (Counter value at external trigger input detection + 1)) × 100

0% output: TKBCR01 setting = 0000H

100% output: TKBCR01 setting ≥ Counter value at external trigger input detection + 1

[Calculation Formula for TKBO01-0, TKBO01-1, TKBO01-2 Output]

Pulse period = (Counter value at external trigger input detection + 1) × Count clock period

Duty [%] = ((Setting value of TKBCR03 – Setting value of TKBCR02) / (Counter value at external trigger input detection + 1)) × 100

0% output: TKBCR03 setting = TKBCR02 setting

100% output: TKBCR02 setting = 0000H

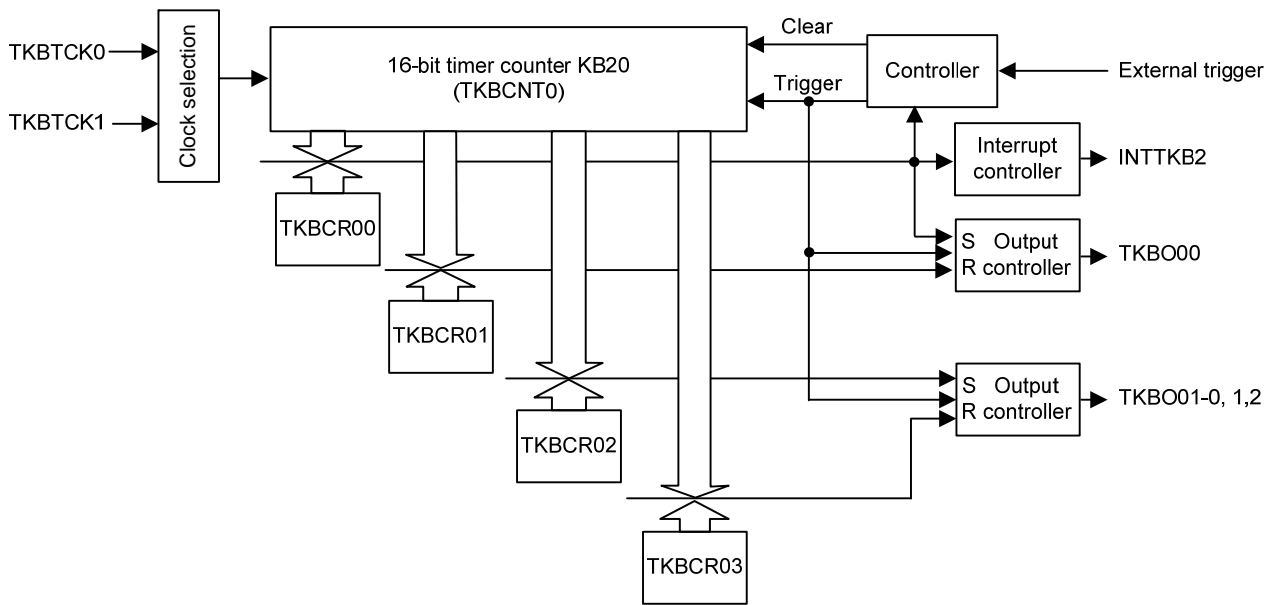
TKBCR03 setting ≥ Counter value at external trigger input detection + 1

**Note** It should always be: TKBCR02 setting ≤ TKBCR03 setting.

Figure 7-44 shows the configuration of standalone mode (period controlled by external trigger input).



**Figure 7-44. Configuration of Standalone Mode (Period Controlled by External Trigger Input)**



**Remark** Output from TKBO01-0, TKBO01-1, and TKBO01-2 is switched by the 16-bit timer KB2 switch register (PWCTKB).

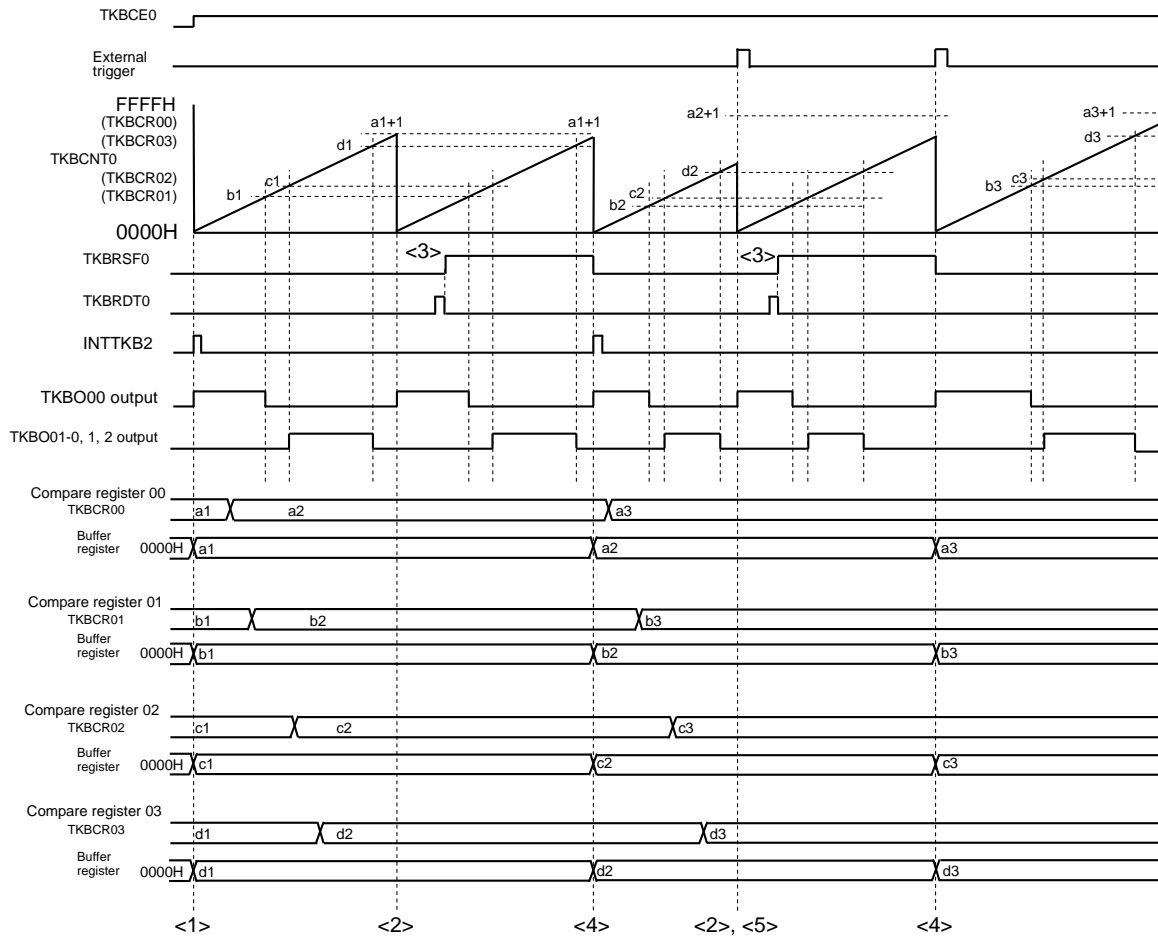
**(2) Batch overwrite function (in standalone operation during period controlled by external trigger input, buffer updating during counting operation (TKBTSE0 bit set to 1))**

In standalone operation during the period controlled by external trigger input, counter clear and compare register batch overwrite are implemented at the timing when external trigger input is detected after writing 1 to the TKBRDT0 bit and by setting the TKBTSE0 bit in TKBCTL00 register to 1. Same as in counter clear, batch overwrite is also implemented when TKBCR00 and counter (TKBCNT0) match before detection of external trigger input after writing 1 to the TKBRDT0 bit. The source of external trigger input is selected by the ELSELR00 to ELSELR09 registers and the TKBSTS01 and TKBSTS00 bits in the TKBCTL00 register.

Figure 7-45 shows an example of the timing of batch overwrite operation when TKBTSE0 bit is set to 1.

- <1> Compare register setting is transferred to the buffer register at the timing when TKBCE0 bit is set from 0 to 1 and TKBCNT0 starts counting operation.
- <2> After the TKBCR00 to TKBCR03 registers are overwritten, even when counter clear is generated, batch overwrite is not generated if 1 is not written to the TKBRDT0 bit.
- <3> The batch overwrite trigger pending status flag (TKBRSF0 bit) is set to 1 by writing 1 to the TKBRDT0 bit.
- <4> When counter clear is generated by external trigger input while TKBTSE0 bit is set to 1 and TKBRSF0 bit is 1, the setting value of the compare register is transferred to the buffer register. At the same time, the TKBRSF0 bit becomes 0. Same as in counter clear, batch overwrite is also implemented when TKBCR00 and counter (TKBCNT0) match before detection of external trigger input after writing 1 to the TKBRDT0 bit.
- <5> Even if the counter clear event is generated by external trigger input, batch overwrite does not occur unless 1 is written to the TKBRDT0 bit.

**Figure 7-45. Batch Overwrite Function: Standalone Operation during Period Controlled by External Trigger Input and Timing of Buffer Updating during Counting Operation (TKBTSE0 bit set to 1)**



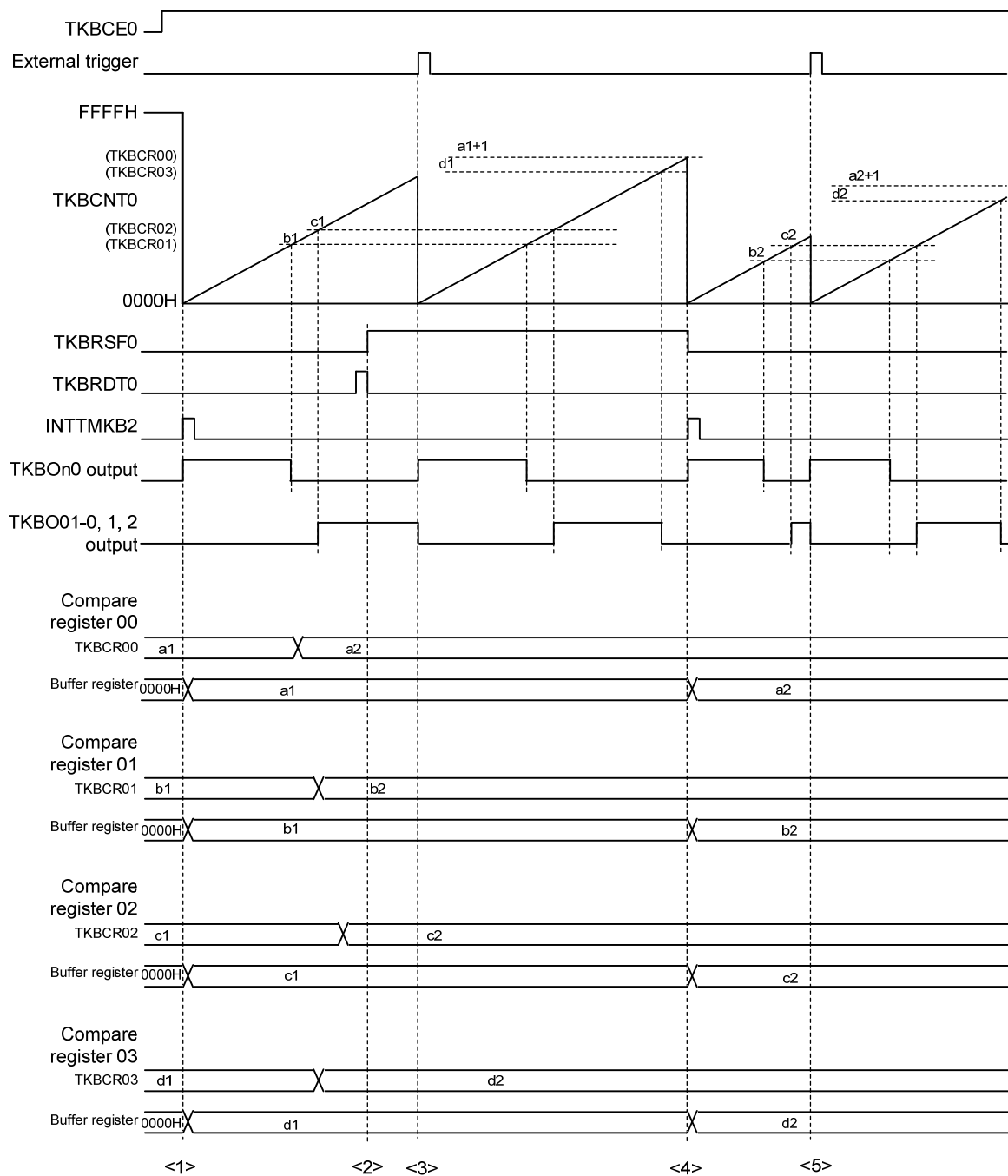
**(3) Batch overwrite function (standalone operation during period controlled by external trigger input, buffer updating during counting operation (TKBTSE0 bit clear to 0))**

This is an example when the TKBTSE0 bit in the TKBCTL00 register is set to 0 in standalone operation during the period controlled by external trigger input. In this case, the counter is cleared when 1 is written to the TKBRDT0 bit and the external trigger input is detected while the batch overwrite trigger pending status flag (TKBRSF0 bit) is 1. However, batch overwrite of the compare register is not implemented.

The source of external trigger input is selected by the ELSELR00 to ELSELR09 registers and the TKBSTS01 and TKBSTS00 bits in the TKBCTL00 register. Figure 7-46 shows an example of the batch overwrite operation timing when the TKBTSE0 bit is set to 0.

- <1> Compare register setting is transferred to the buffer register at the timing when the TKBCE0 bit is set from 0 to 1 and TKBCNT0 starts counting operation.
- <2> After the TKBCR00 to TKBCR03 registers are overwritten, the batch overwrite trigger pending status flag (TKBRSF0 bit) is set to 1 by writing 1 to the TKBRDT0 bit.
- <3> Even if the counter clear event is generated by external trigger input, batch overwrite does not occur unless the TKBTSE0 bit is set to 1.
- <4> When the counter clear event (TKBCNT0 and TKBCR00 match) is generated while TKBRSF0 bit is 1, the setting value of the compare register is transferred to the buffer register. At the same time, the TKBRSF0 bit becomes 0.
- <5> Even if the counter clear event is generated by external trigger input, batch overwrite does not occur unless the TKBTSE0 and TKBRSF0 bits are both 1.

**Figure 7-46. Batch Overwrite Function: Standalone Operation during Period Controlled by External Trigger Input and Timing of Buffer Updating during Counting Operation (TKBTSE0 bit cleared to 0)**



(4) Sample of register setting details at standalone mode (period controlled by external trigger input)

	15	14	13	12	11	10	9	8
TKBCTL00	TKBIHE0 0	– 0	TKBSSE01 0	TKBDIE01 0	– 0	– 0	TKBSSE00 0	TKBDIE00 0
	7	6	5	4	3	2	1	0
	TKBMFE0 1/0	– 0	TKBIRS01 0	TKBIRS00 0	– 0	TKBTSE0 1/0	TKBSTS01 1/0	TKBSTS00 1/0
TKBCTL01	TKBCE0 1	– 0	– 0	TKBCKS0 1/0	– 0	– 0	TKBMD01 0	TKBMD00 0
TKBIOC00	– 0	– 0	– 0	– 0	TKBTOL01 1/0	TKBTOL00 1/0	TKBTOD01 1/0	TKBTOD00 1/0
TKBIOC01	TKBNFB0 0	– 0	TKBEGPA0 0	TKBEGNA0 0	TKBEGPB0 0	TKBEGNB0 0	TKBTOE01 1/0	TKBTOE00 1/0
TKBPSCS0	– 0	TKBTPS012 1/0	TKBTPS011 1/0	TKBTPS010 1/0	– 0	TKBTPS002 1/0	TKBTPS001 1/0	TKBTPS000 1/0
TKBCR00	0000H to FFFFH							
TKBCR01	0000H to FFFFH							
TKBCR02	0000H to FFFFH							
TKBCR03	0000H to FFFFH							
TKBSIR00	0000H							
TKBSIR01	0000H							
TKBSSR00	00H							
TKBSSR01	00H							
TKBDNR00	00H							
TKBDNR01	00H							
TKBMFR0	0000H to FFFFH							

☐ : Setting is fixed for this mode      ■ : Setting is not needed (default setting)

### 7.4.7 Interleave PFC (Power Factor Correction) output mode

This is the mode that can generate a signal as interleave output that controls PFC circuit which regulates the harmonic current of the power source.

As interleaved PFC circuit can regulate peak input current at greater extent than single PFC circuit, it can make parts smaller and implement high powered power source units.

Interleaved PFC control requires two inputs for zero current detection and two PWM outputs for switching.

Interleaved PFC output mode is implemented by a combination of external interrupt input selected for counter restart trigger source 0 and TKBO00, and external interrupt input selected for counter restart trigger source 1 and TKBO01-0, TKBO01-1, TKBO01-2.

The TKBO01-0, TKBO01-1, and TKBO01-2 phases shifted by 180 degrees by external interrupt input selected for counter restart trigger source 1 are output based on the TKBO00 output controlled by external interrupt input selected for counter restart trigger source 0.

**Remark** Single PFC control can be implemented in standalone mode (period controlled by external input trigger). For more details, see **7.4.6 Standalone mode (period controlled by external trigger input)**.

The counter restart period is set by TKBCR00 if external interrupt input selected for counter restart trigger source 0 is not detected.

The active width of TKBO00 output is set by TKBCR01.

The active width of TKBO01-0, TKBO01-1, TKBO01-2 output is set by TKBCR03.

**Remark** Interleave PFC (Power Factor Correction) output mode does not use TKBCR02.

The setting value of the TKBTOL00 and the TKBTOD00 bit, and the TKBTOL01 bit and the TKBTOD01 bit must be the same value. This makes that when the default level is low (high) level, the active level becomes high (low) level.

[Calculation Formula for TKBO00 Output and TKBO01-0, TKBO01-1, TKBO01-2 Output]

Pulse period (MAX)<sup>Note</sup> = (TKBCR00 setting + 1) × Count clock period

Active width of TKBO00 output = TKBCR01 setting × Count clock period

Active width of TKBO01-0, TKBO01-1, TKBO01-2 output = TKBCR03 setting × Count clock period

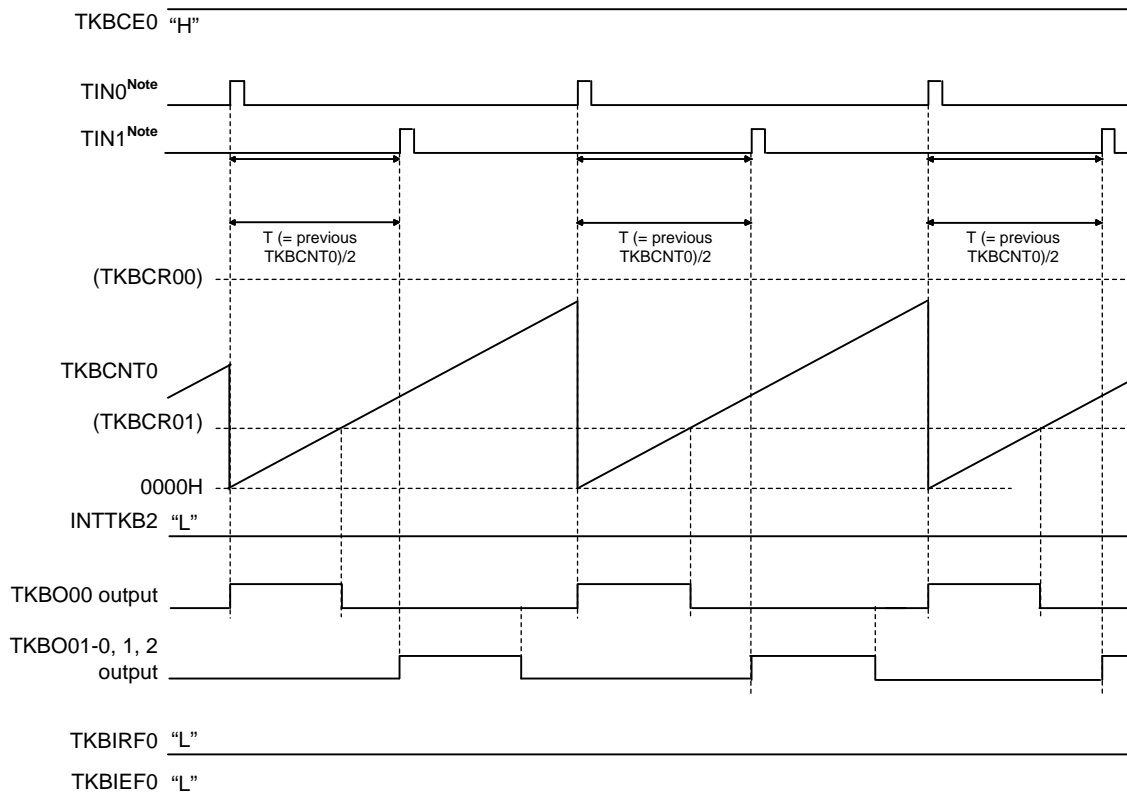
**Note** This is the counter restart period in case when external interrupt input selected for counter restart trigger source 0 not being detected.

Figure 7-47 shows the overview of basic operation of interleave PFC mode. In basic operation of interleave PFC mode, TKBCNT0 is incremented from 0000H by external interrupt input selected for counter restart trigger source 0 as a trigger. In this case, TKBO00 becomes active level, and then becomes inactive level when it matches the setting value of the TKBCR01 register.

TKBO01-0, TKBO01-1, TKBO01-2 becomes active level by being triggered by external interrupt input selected for counter restart trigger source 0, and becomes inactive level when it matches the setting value of TKBCR03 register.

Another external interrupt input selected for counter restart trigger source 0 comes in before TKBCNT0 matches the setting value of the TKBCR00 register, and then the above operation is repeated.

**Figure 7-47. Overview of Basic Operation of Interleave PFC Mode**  
 (When default value of output is low level (TKBTOD0p = 0) and active level is high level (TKBTOL0p = 0))



**Note** TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

**Remark** p = 0, 1



**(1) Output conditions of TKBO01-0, TKBO01-1, TKBO01-2 at interleave PFC**

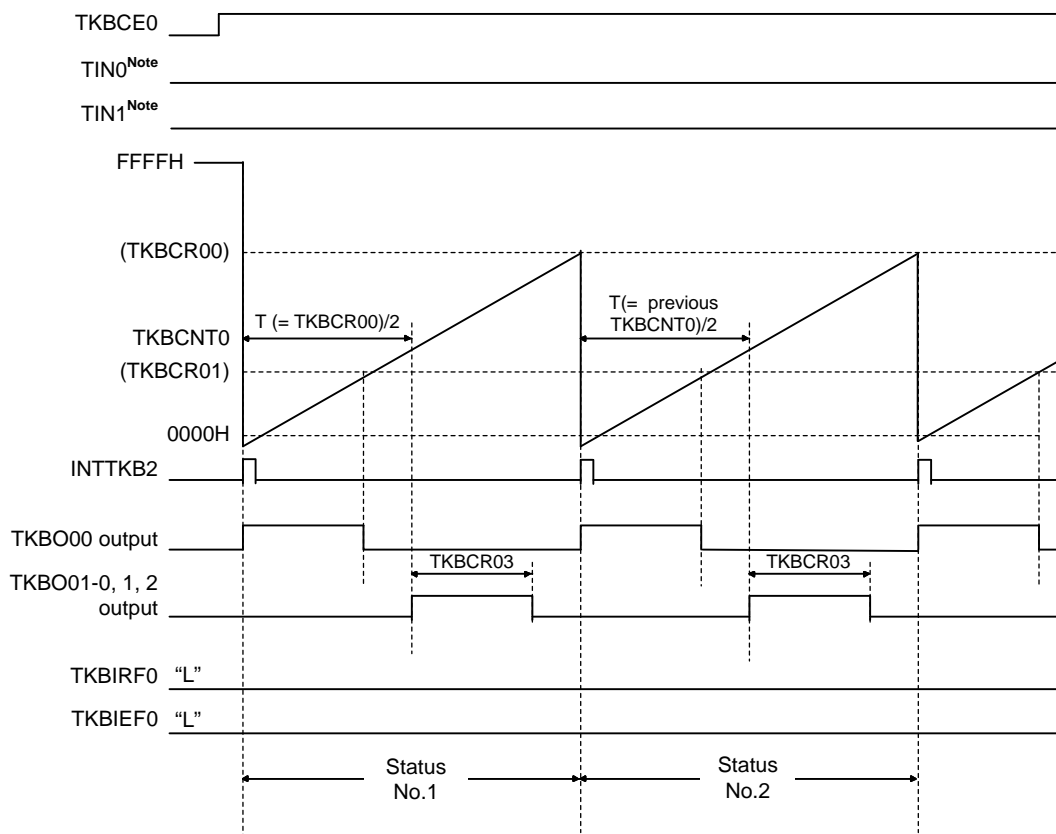
There are output conditions for TKBO01-0, TKBO01-1, TKBO01-2 output which are controlled according to the table below.

Status No.	Output Conditions for TKBO01-0, TKBO01-1, TKBO01-2 Output			TKBO01-0, TKBO01-1, TKBO01-2 Start at High Level
	TIN0 <sup>Note 1</sup> Input	Matching with CR00/TIN1 <sup>Note 1</sup>	Period Width	
1	First period	–	–	Output start by T/2 (CR00 setting is T)
2	TIN0 input not detected	Matching of TKBCNT0 and CR00 (Ignore TIN1 input detection)	Subsequent period (CR00 value) is over 1/2 the previous period	Output start by T/2
3	↑	↑	Subsequent period (CR00 value) is below 1/2 the previous period	Maintain the status
4	Subsequent period of No.3	–	–	Output start by T/2
5	TIN0 input detected (for the first time) <sup>Note 2</sup>	–	–	Output start by T/2
6	TIN0 input detected (from the second time) <sup>Note 3</sup>	TIN1 detected (within the range from previous TOUT1 falling edge to T/2)	–	Output start by T/2
7	TIN0 input detected (from the second time) <sup>Note 3</sup>	TIN1 detected (T/2 to T/2 + T/ (TKBIRS01 and TKBIRS00 setting) range)	–	Output start by trigger input
8	TIN0 input detected (from the second time) <sup>Note 3</sup>	TIN1 detected (after the range T/2 + T/(IRS1 and IRS0 setting)	–	Maintain the status
9	Subsequent period of No.8	–	–	Output start by T/2
10	TIN0 input detected	–	Subsequent period is below T/2	Maintain the status
11	Subsequent period of No.10	–	–	Output start by T/2

- Notes**
1. TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.
  2. TIN0 input detected (for the first time) means that the previous period was not cleared for TIN0 input being detected.
  3. TIN0 input detected (from the second time) means that the previous period being cleared for TIN0 input being detected.

See the following figures of the waveform corresponding to each “Condition No.”

**Figure 7-48. Timing of Interleave PFC Mode (Operation for Status No. 1 and No. 2)**

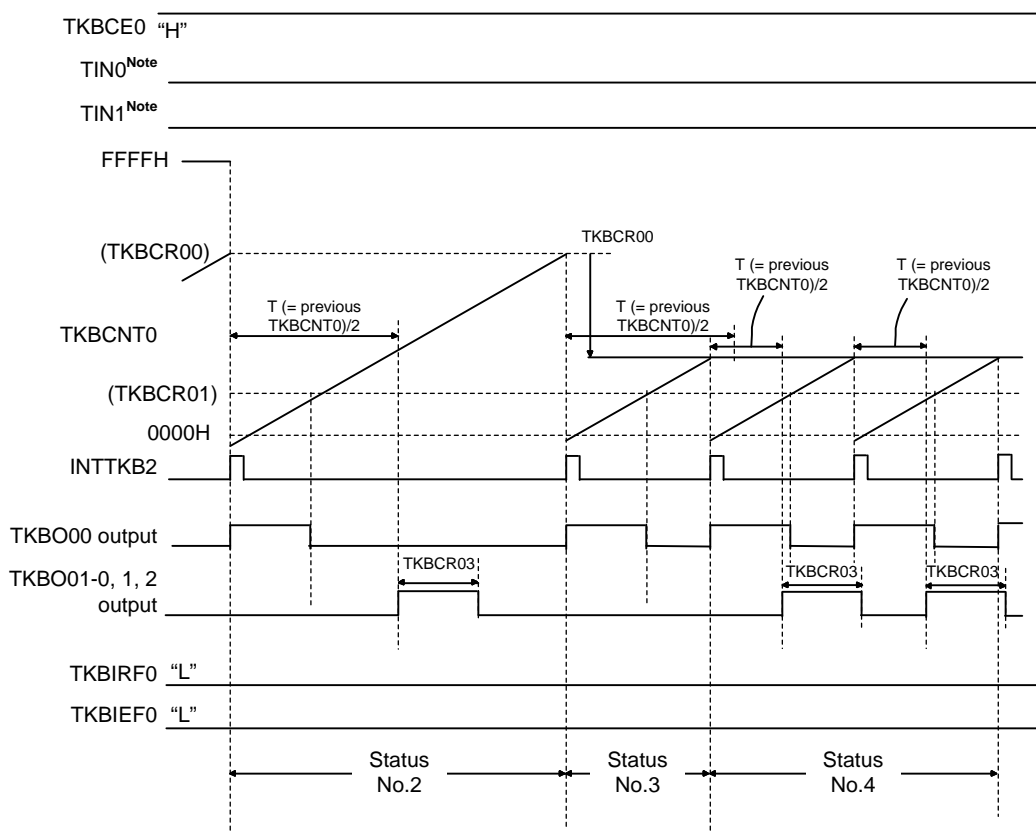


**Note** TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

**Status No.1:** Only for the first period after  $\text{TKBCE0} = 1$  setting, TKBO1-0, TKBO1-1, TKBO1-2 with setting width of TKBCR03 is output setting "T" as TKBCR00.

**Status No.2:** In the second period, TKBO1-0, TKBO1-1, TKBO1-2 with setting width of TKBCR03 is output at T/2 of the previous period.

**Figure 7-49. Timing of Interleave PFC Mode (Operation from Status No. 3 and No. 4)**  
 (When default value of output is low level (TKBTOD0p = 0) and active level is high level (TKBTOL0p = 0))



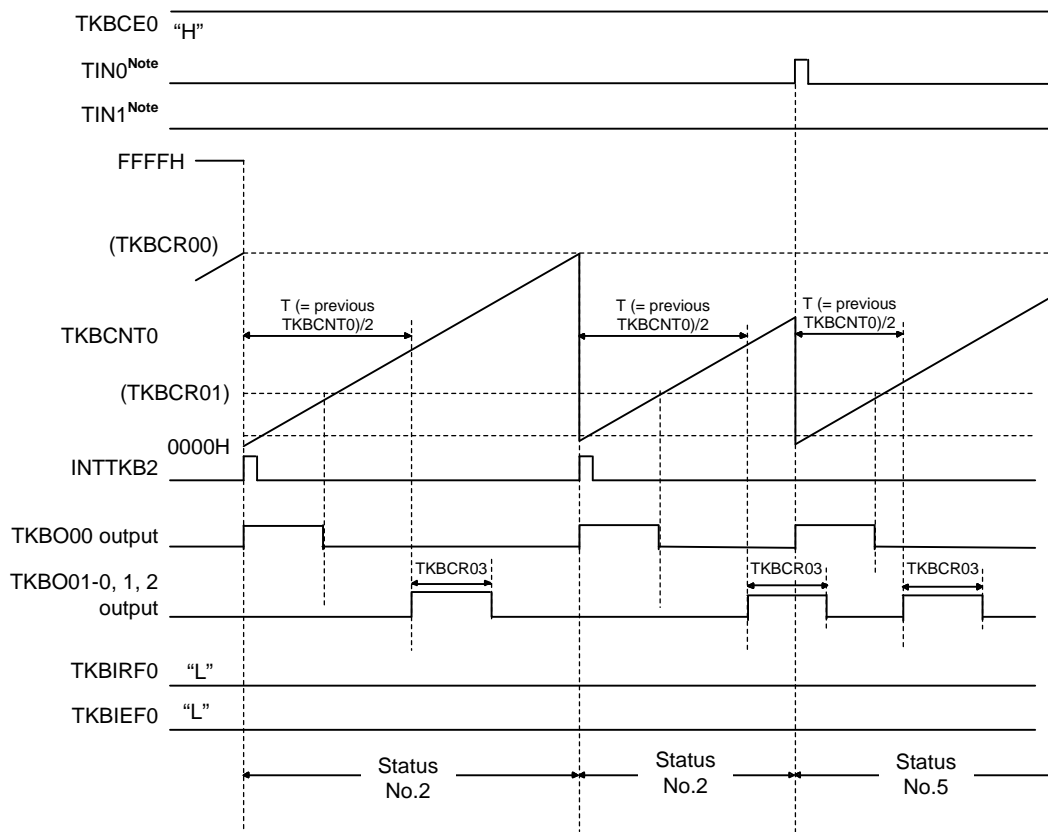
**Note** TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

**Remark** p = 0, 1

**Status No.3:** TKBO01-0, TKBO01-1, TKBO01-2 maintains the status and T/2 of the previous period is not ensured.

**Status No.4:** TKBO01-0, TKBO01-1, TKBO01-2 with setting width of TKBCR03 is output at T/2 of the previous period.

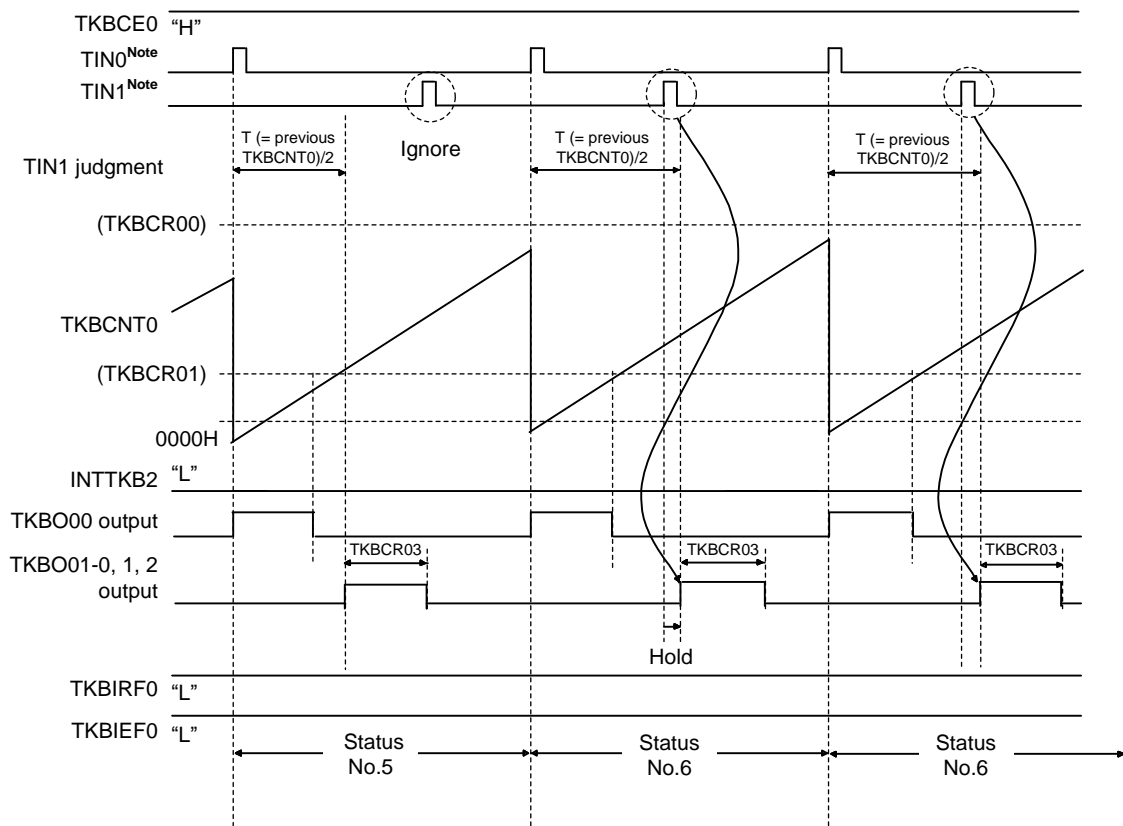
**Figure 7-50. Timing of Interleave PFC Mode  
(Operation for Status No. 5: INT0 Input Detected (for the first time))**



**Note** TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

**Status No.5:** TIN0 which was first detected after setting  $TKBCE0 = 1$  outputs TKBO01-0, TKBO01-1, TKBO01-2 with setting width of TKBCR03 at  $T/2$  of the previous period. It does not depend on whether TIN1 is detected or not detected.

**Figure 7-51. Timing of Interleave PFC Mode (Operation for Status No. 6)**  
 (When default value of output is low level (TKBTOD<sub>p</sub> = 0) and active level is high level (TKBTOL<sub>p</sub> = 0))

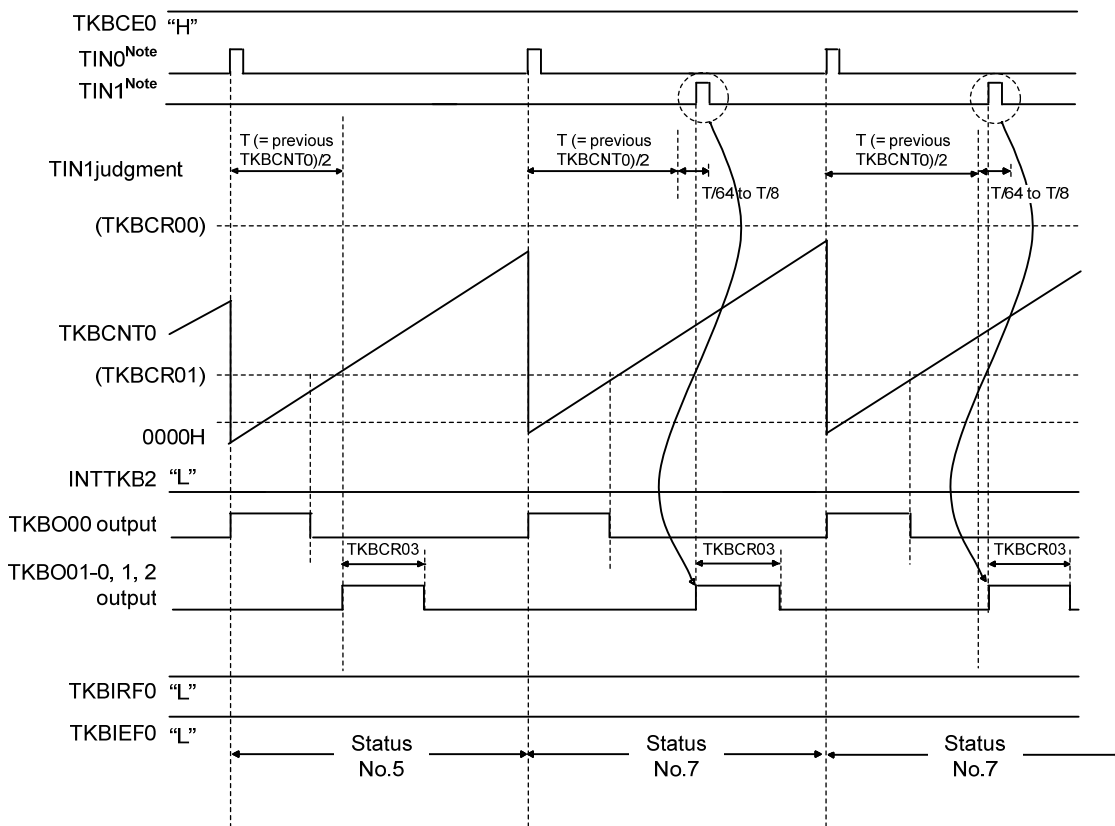


**Note** TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

**Remark**  $p = 0, 1$

**Status No.6:** TKBO01-0, TKBO01-1, TKBO01-2 with setting width of TKBCR03 is output at T/2 of the previous period as TIN1 input is below T/2 of the previous period.

**Figure 7-52. Timing of Interleave PFC Output Mode (Operation for Status No. 7)**  
 (When default value of output is low level (TKBTOD0p = 0) and active level is high level (TKBTOL0p = 0))

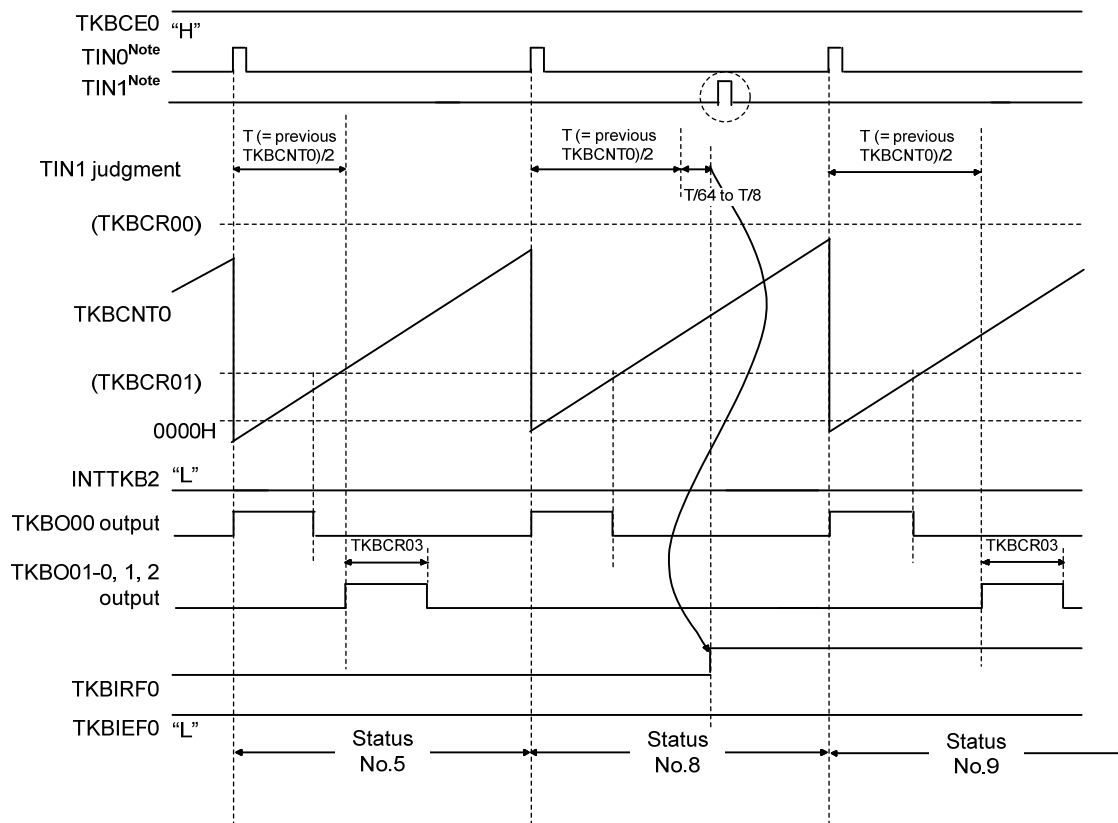


**Note** TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

**Remark** p = 0, 1

**Status No.7:** After the detection of TIN0 when TIN1 is detected over  $T/2$  of the previous period and within  $T/2 + T/n$  ( $n = 8, 16, 32, 64$ : set by TKBIRS01 and TKBIRS00), TKBO01-0, TKBO01-1, TKBO01-2 with setting width of TKBCR03 is output at the detection of TIN1.

**Figure 7-53. Timing of Interleave PFC Output Mode (Operation for Status No. 8 and No. 9)**  
 (When default value of output is low level (TKBTOD0p = 0) and active level is high level (TKBTOL0p = 0))



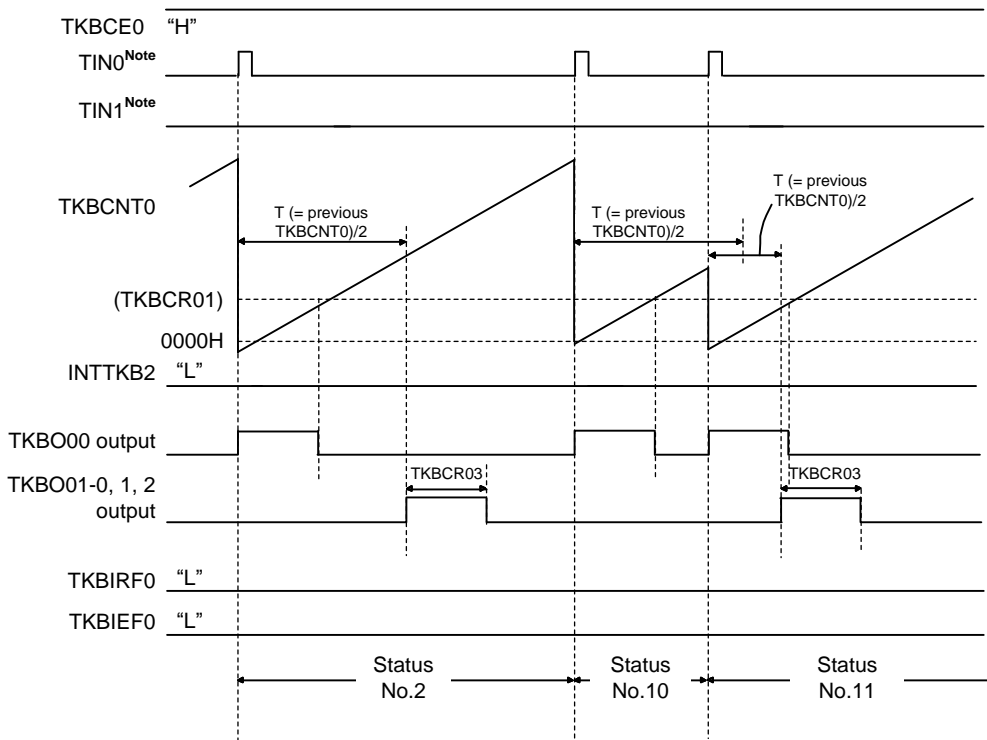
**Note** TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

**Remark** p = 0, 1

**Status No.8:** If TIN1 is not detected within  $T/2 + T/n$  ( $n = 8, 16, 32, 64$ : set by TKBIRS01 and TKBIRS00) of the previous period, TKBO01-0, TKBO01-1, TKBO01-2 maintains the status. TKBIRF0 is set to 1 at this time.

**Status No.9:** TKBO01-0, TKBO01-1, TKBO01-2 with setting width of TKBCR03 is output at  $T/2$  of the previous period.

**Figure 7-54. Timing of Interleave PFC Output Mode (Operation for Status No. 10 and No. 11)**  
 (When default value of output is low level (TKBTOD0p = 0) and active level is high level (TKBTOL0p = 0))



**Note** TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

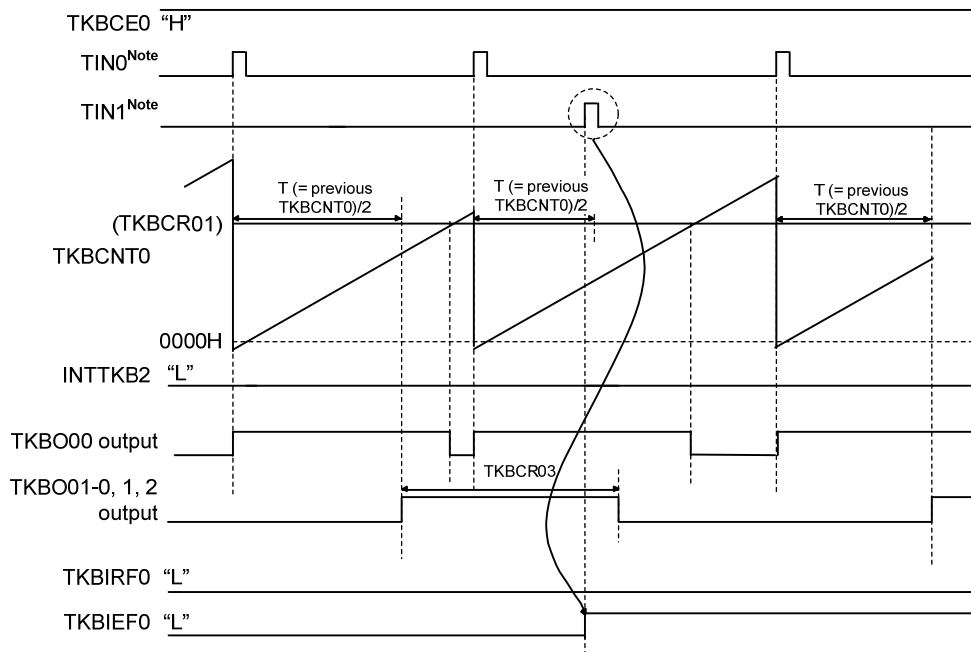
**Remark** p = 0, 1

**Status No.10:** TKBO01-0, TKBO01-1, TKBO01-2 maintains the status and T/2 of the previous period is not ensured.

**Status No.11:** TKBO01-0, TKBO01-1, TKBO01-2 with setting width of TKBCR03 is output at T/2 of the previous period.



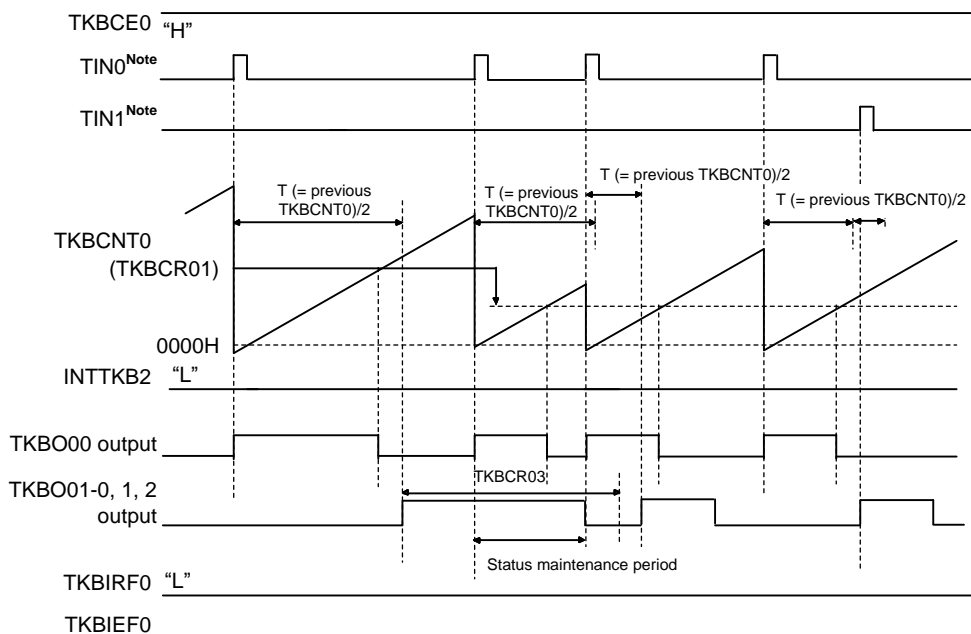
**Figure 7-55. Timing of Interleave PFC Output Mode**  
 (When trigger is generated again while TKBO01-0, TKBO01-1, and TKBO01-2 are output)



**Note** TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

The trigger is ignored when the subsequent TKBO01-0, TKBO01-1, TKBO01-2 output trigger is generated while outputting TKBO01-0, TKBO01-1, TKBO01-2 of the previous period. TKBIEF0 is set to 1 at this time.

**Figure 7-56. Timing of Interleave PFC Output Mode**  
 (TKBO01-0, TKBO01-1, TKBO01-2 output is at the width of the previous output width and exceeds status maintenance period)



**Note** TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

When TKBO01-0, TKBO01-1, TKBO01-2 output of the previous output width is long which exceeds status maintenance period, it is default output compulsively at the starting timing of the subsequent period following the completion of the status maintenance period.

(2) List of register setting at interleave PFC output mode

	15	14	13	12	11	10	9	8
TKBCTL00	TKBIHE0 0	– 0	TKBSSE01 0	TKBDIE01 0	– 0	– 0	TKBSSE00 0	TKBDIE00 0
	7	6	5	4	3	2	1	0
	TKBMFE0 1/0	– 0	TKBIRS01 1/0	TKBIRS00 1/0	– 0	TKBTSE0 1	TKBSTS01 0	TKBSTS00 0
TKBCTL01	TKBCE0 1	– 0	– 0	TKBCKS0 1/0	– 0	– 0	TKBMD01 1	TKBMD00 1
TKBIOC00	– 0	– 0	– 0	– 0	TKBTOL01 1/0	TKBTOL00 1/0	TKBTOD01 1/0	TKBTOD00 1/0
TKBIOC01	TKBNFB0 0	– 0	TKBEGPA0 0	TKBEGNA0 0	TKBEGPB0 0	TKBEGNB0 0	TKBTOE01 1/0	TKBTOE00 1/0
TKBPSCS0	– 0	TKBTPS012 1/0	TKBTPS011 1/0	TKBTPS010 1/0	– 0	TKBTPS002 1/0	TKBTPS001 1/0	TKBTPS000 1/0
TKBCR00	0000H to FFFFH							
TKBCR01	0000H to FFFFH							
TKBCR02	0000H to FFFFH							
TKBCR03	0000H to FFFFH							
TKBSIR00	0000H							
TKBSIR01	0000H							
TKBSSR00	00H							
TKBSSR01	00H							
TKBDNR00	00H							
TKBDNR01	00H							
TKBMFR0	0000H to FFFFH							

: Setting is fixed for this mode     : Setting is not needed (default setting)

### 7.5 Option Functions of 16-bit Timer KB20

Option functions can be added to timer KB20.

The following table shows available options for each operation mode for timer KB20.

Operation Mode		Standalone Mode		Interleave PFC Output Mode
Period Controlling Method for Operation Mode		Period Controlled by CR00	Period Controlled by Trigger	Period Controlled by Restart Trigger Source 0/CR00
Optional Functions	PWM output dithering function	√ <sup>Note 1</sup>	–	–
	PWM output smooth start function	√ <sup>Note 1</sup>	–	–
	Maximum frequency limit function	–	√ <sup>Note 2</sup>	√
	PWM output function for IH control	√	√	–

- Notes**
- Do not use when using the PWM output function for IH control.
  - Restart trigger input of PWM output function for IH control is not supported for detecting maximum frequency limit function.

**Remark** For details of the operation specifications, see **7.4.2 Default level and active level** and **7.4.3 Stop/restart operation**.

#### 7.5.1 PWM output dithering function

16-bit timer KB20 is available for high resolution PWM using PWM output dithering function.

Taking 16 periods of the PWM period as one unit, 16 times higher PWM is available for average resolution by extending the active period of N times (N = 0 to 15) by one count clock during one unit.

The number of repetitions (N) extending the active period by one count clock during one unit is set to the TKBDNR0p register. The following figure shows the ordinal of the period (kth period) extending the active period (N times) by one count clock during one unit.

For example, when N = 3, the PWM active period is extended by one clock at the first, fifth, and ninth periods during one unit.

**Figure 7-57. 16-bit Timer KB2 Dithering Count Register 0p (TKBDNR0p) Setting**

kth period Repetitions (N)	k															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0																
1	■															
2	■							■								
3	■				■			■								
4	■				■			■				■				
5	■		■		■			■				■				
6	■		■		■			■			■					
7	■		■		■		■				■					
8	■		■		■		■				■				■	
9	■		■		■		■				■				■	
10	■		■		■		■			■					■	
11	■		■		■		■			■					■	
12	■		■		■		■			■					■	
13	■		■		■		■			■					■	
14	■		■		■		■			■					■	
15	■		■		■		■			■					■	

- Remarks**
- cell: Set to active period according to settings in TKBCR01 and TKBCR03 registers
    - cell: Set to active period according to “settings + 1” in TKB0CR1 and TKBCR03 registers
  - p = 0, 1

Figure 7-58. Waveform at Dithering Operation

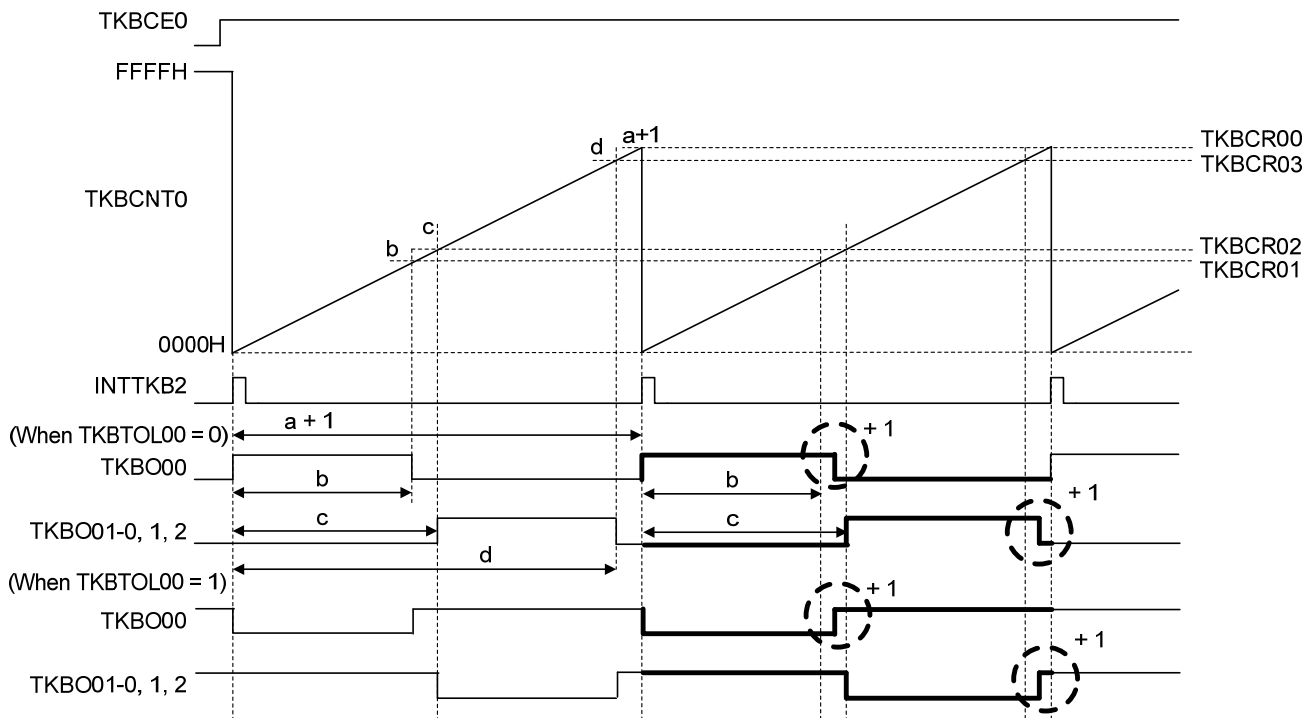
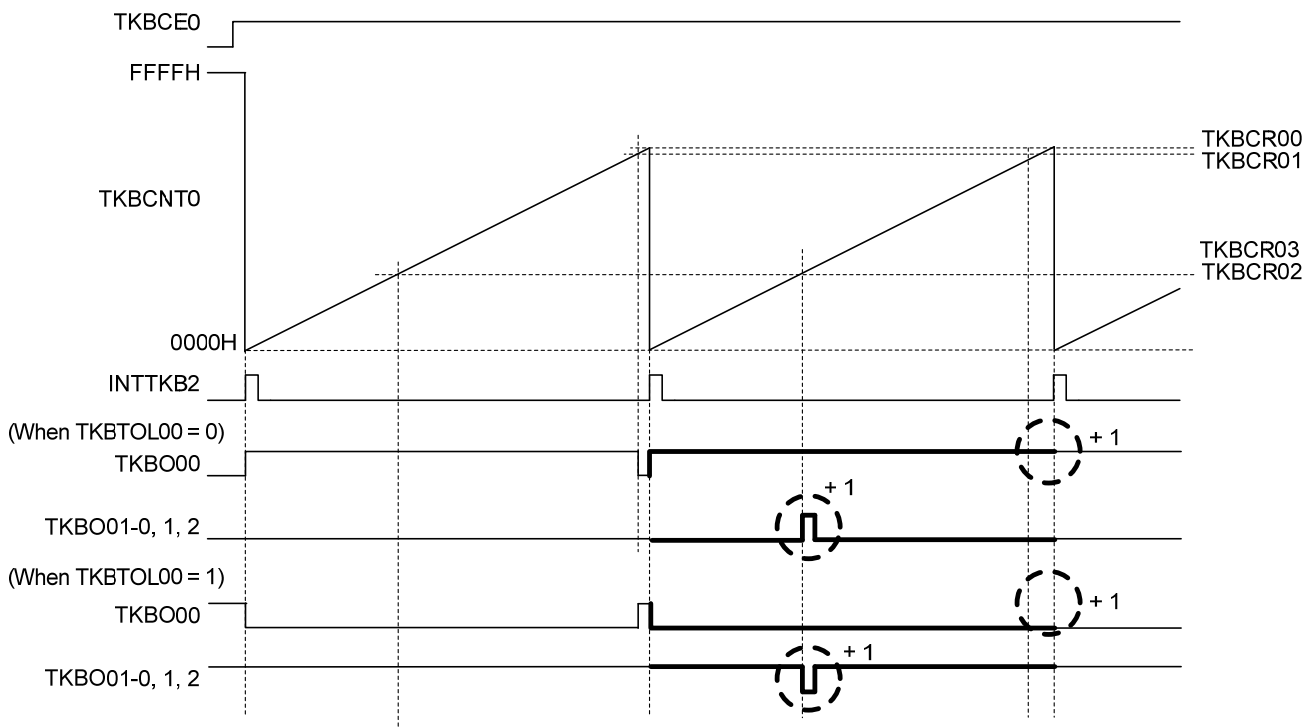
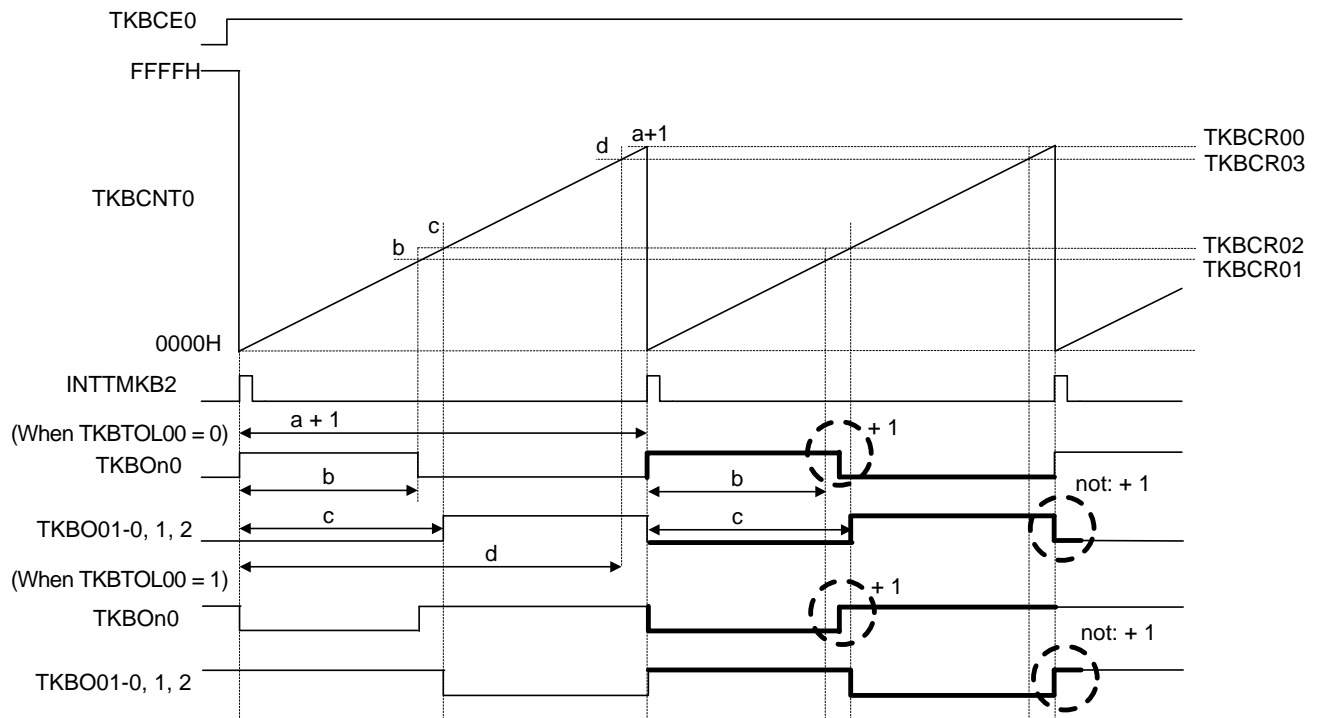


Figure 7-59. Waveform at Dithering Operation  
(When  $TKBCR01 = TKBCR00$  (100% nearest neighbor),  $TKBCR02 = TKBCR03$  (0% nearest neighbor))



**Figure 7-60. Waveform at Dithering Operation  
(When  $TKBCR03 = TKBCR00 + 1$ )**



**(1) Available operation mode**

The following shows enable or disable status under each mode that is specified by the TKBCTL00 register (TKBSTS01 and TKBSTS00 bits) and the TKBCTL01 register (TKBMD01 and TKBMD00 bits).

Operation Mode	TKBMD01, TKBMD00	TKBSTS01, TKBSTS00	Setting
Standalone mode (Period controlled by TKBCR00)	00B	00B	Available
Standalone mode (Period controlled by external trigger input)	00B	01B/10B/11B	Not available
Interleave PFC output mode	11B	–	Not available

PWM output dithering function is available when external trigger input is not used and the period being controlled by TKBCR00.

TKBDNR00/TKBDNR01 controls PWM output dithering function of respective TKBO00/TKBO01-0, TKBO01-1, TKBO01-2.

**Cautions 1. [Overwrite during Operation (TKBCE0 = 1) of TKBDNR00/TKBDNR01 Register]**

Since TKBDNR00/TKBDNR01 owns the buffer, overwrite during the operation (TKBCE0 = 1) is available.

At this time, batch overwriting is available via writing 1 to the TKBRDT0 bit.

**2. [Access by TKBCRLD00/TKBCRLD01 Register]**

TKBCRLD00 is a 16-bit register mapping lower 8-bit TKBCR01 and TKBDNR00.

TKBCRLD01 is a 16-bit register mapping lower 8-bit TKBCR03 and TKBDNR01.

The value of TKBDNR00/TKBDNR01 is changed even when the TKBCRLD00/TKBCRLD01 register is accessed.

The value of TKBCR01/TKBCR03 is changed even when the TKBCRLD00/TKBCRLD01 register is accessed.

Only the lower 8 bits of TKBCR01/TKBCR03 are changed when the TKBCRLD00/TKBCRLD01 register is accessed.

**3. [To Combine PWM Output Smooth Start Function with PWM Output Dithering Function]**

PWM output dithering function is invalid during the execution of PWM output smooth start function (TKBSSF0p = 1).

PWM output dithering function is valid when PWM output smooth start function is stopped (TKBSSF0p = 0).

**7.5.2 PWM output smooth start function**

Timer KB20 has PMW output smooth start function corresponding to rush current control and over-voltage prevention. PWM output smooth start function begins at timer start timing. The process that a user has performed with software in the past can be easily accomplished with the optional function of the hardware.

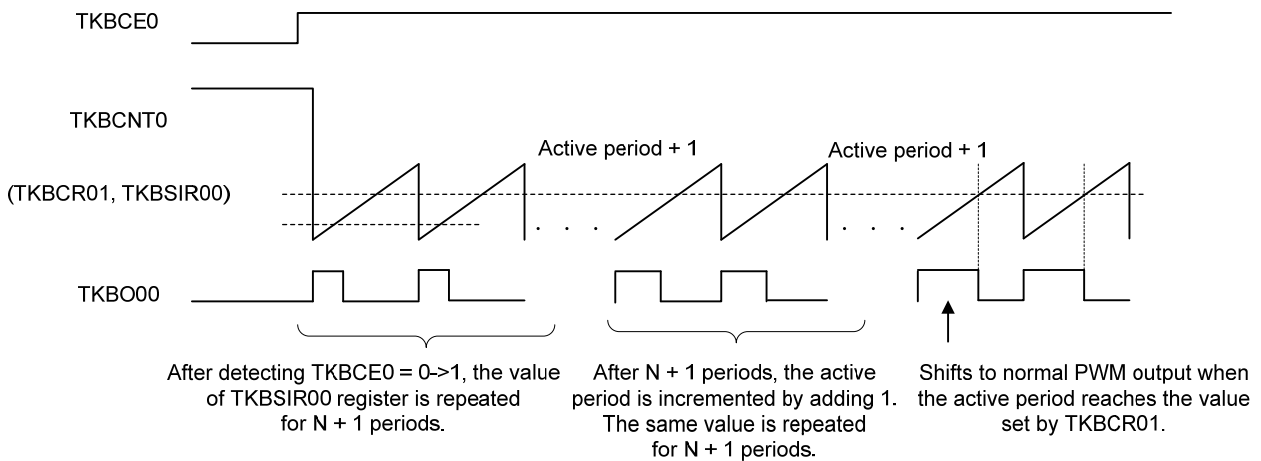
Operation starts with the value set by the 16-bit timer KB2 smooth start default duty register (TKBSIR0p) and the PWM active period is sequentially incremented by one clock. The rate to increment the active period is specified by the 16-bit timer KB2 smooth start step width register (TKBSSR0p). When the value set by the TKBSSR0p register is N, after the currently set active period is output for N + 1 times, the active period is incremented by adding 1 and then the new active period is output for N + 1 times. After repeating this operation to increment the active period, PWM output smooth start function is cancelled when the same active period specified by the TKBCR01 and TKBCR03 registers is reached.

The 16-bit timer KB2 smooth start default duty register should be set according to the following condition;

$$0000H \leq TKBSIR00 < TKBCR01 \leq TKBCR00 + 1$$

$$TKBCR02 \leq TKBSIR01 < TKBCR03 \leq TKBCR00 + 1$$

**Figure 7-61. Example of TKBO00 Output Using PWM Output Smooth Start Function**



- Remarks 1.** N: Value set by  $TKBSSR0p$  register
- 2.**  $p = 1, 0$

**(1) Operation mode available for PMW output smooth start function**

Operation Mode	TKBMD01, TKBMD00	TKBSTS01, TKBSTS00	Setting
Standalone mode (Period controlled by $TKBCR00$ )	00B	00B	Available
Standalone mode (Period controlled by external trigger input)	00B	01B/10B/11B	Not Available
Interleave PFC output mode	11B	–	Not Available



**(2) Overwrite during operation (TKBCE0 = 1) of TKBSIR00/TKBSIR01/TKBSSR00/TKBSSR01 registers**

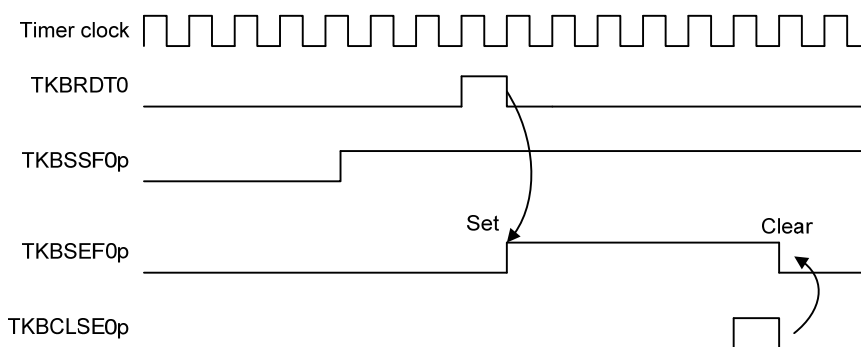
Overwrite during the operation (TKBCE0 = 1) is available for TKBSIR00/TKBSIR01/TKBSSR00/TKBSSR01. TKBSIR00/TKBSIR01/TKBSSR00/TKBSSR01 own the buffer and batch overwriting is available via writing 1 to the TKBRDT0 bit. When restarting the smooth start function, clear the TKBCE0 bit to 0, and then set it to 1.

**(3) Overwrite during operation (TKBCE0 = 1) of**

**TKBCR00/TKBCR01/TKBCR02/TKBCR03/TKBSIR00/TKBSIR01/TKBSSR00/TKBSSR01 registers**

When TKBRDT0 is set to 1 during the period of PWM output smooth start (TKBSSF00 = 1, TKBSSF01 = 1), batch overwrite is masked and the TKBSEF0p flag is set. In order to perform batch overwrite, clear TKBSEF0p and confirm TKBSSF0p becomes 0, then set TKBRDT0 to 1.

**Figure 7-62. Overwrite During Smooth Start Function Operation (TKBSSF0p = 1) of TKBCR00/TKBCR01/TKBCR02/TKBCR03/TKBSIR00/TKBSIR01/TKBSSR00/TKBSSR01 Registers**



**Remark** p = 1, 0

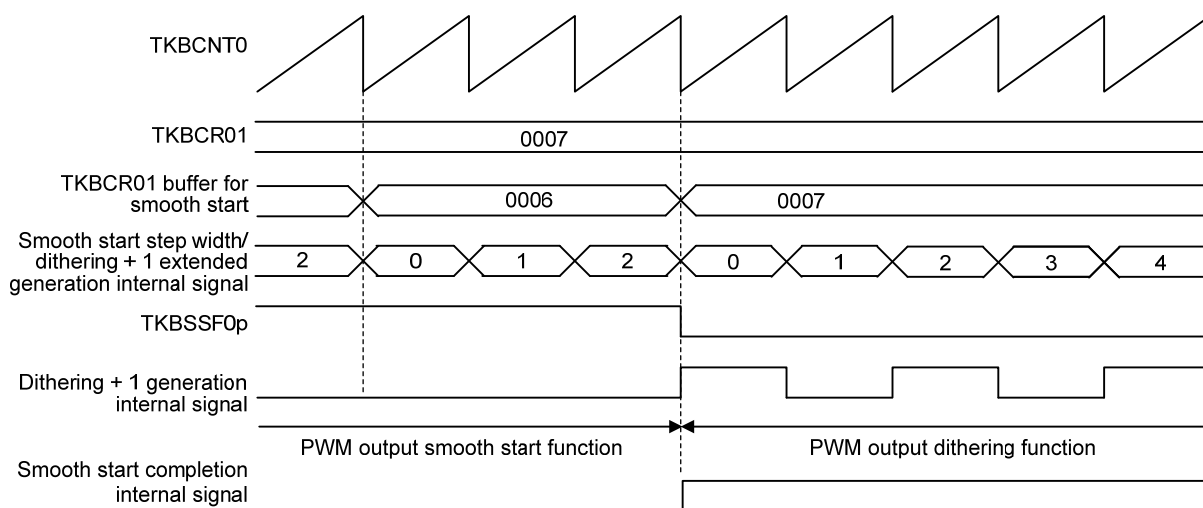
**(4) To combine PWM output smooth start function with PWM output dithering function**

PWM output dithering function is invalid during the execution of PWM output smooth start function (TKBSSF0p = 1). PWM output dithering function will be valid when PWM output smooth start function is stopped (TKBSSF0p = 0).

**(5) Completion of PWM output smooth start function and operation of TKBSSF0p**

The following figure shows when TKBCR01 is 0007H, TKBDNR0p is 70H and TKBSSR0p is 02H. At the timing that TKBCR01 = 0007H and the value of TKBCR01 buffer for internal smooth start matches, TKBSSF0p is cleared, and then dithering function begins.

**Figure 7-63. Completion of PWM Output Smooth Start Function and Operation of TKBSSF0p**



### 7.5.3 Maximum frequency limit function

Timer KB20 has a function that regulates the minimum period of the counter clear (maximum frequency) in the periodic control by external trigger or interleave PFC output mode.

When this function is used, if external trigger input which performs the counter clear occurs while the counter value is less than the setting value of the maximum frequency limit register (TKBMFR0), it performs the counter clear after it continues counting until it reaches the setting value of the TKBMFR0 register.

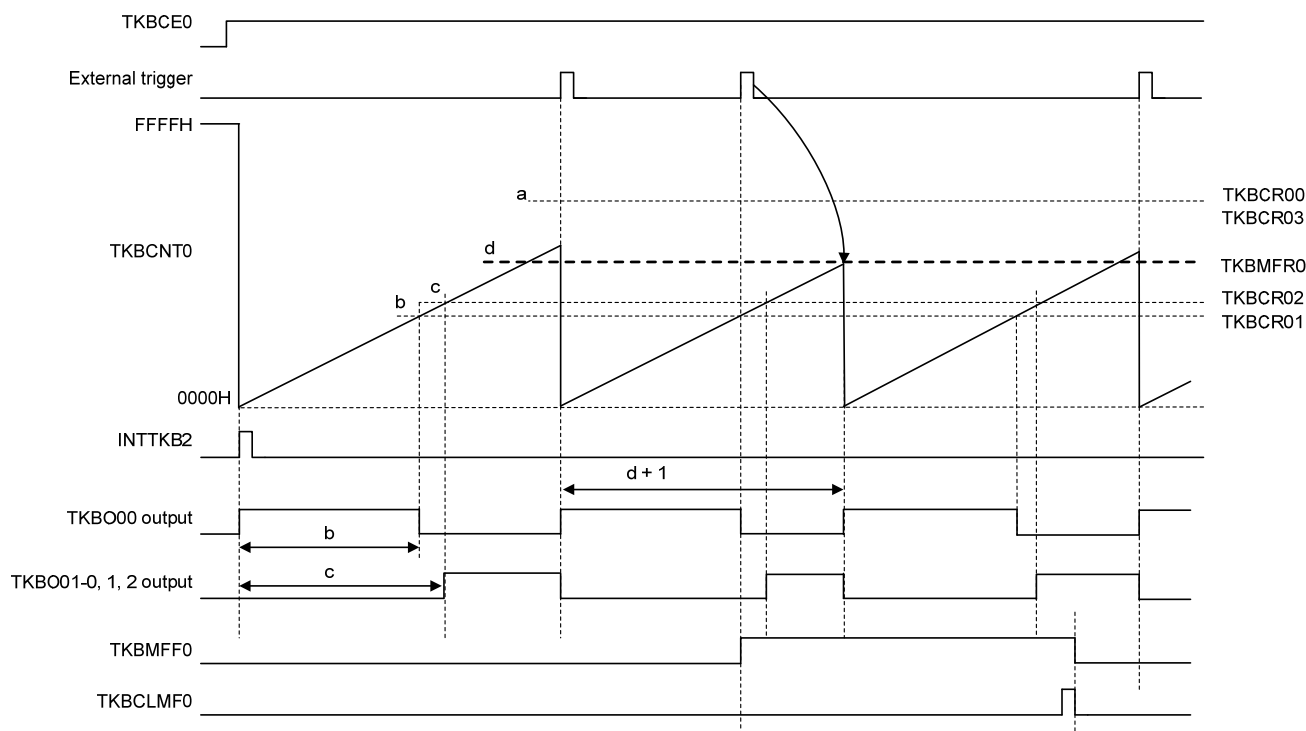
#### (1) Formula for maximum frequency limit (= 1/minimum period)

$$\text{Minimum period} (= 1/\text{Maximum frequency limit}) = (\text{TKBMFR0 setting} + 1) \times \text{Count clock period}$$

**Note** The following condition must be satisfied: TKBMFR0 setting ≤ TKBCR00 setting

When the counter value is smaller than TKBMFR0 at the timing for external trigger input detection, the TKBMFF0 flag is set to 1. The TKBMFF0 flag is cleared to 0 by writing 1 to the TKBCLMF0 bit.

Figure 7-64. Maximum Frequency Limit Function



**Remark** Period controlled by external trigger input.

#### (2) Operation mode available for maximum frequency limit function

Operation Mode	TKBMD01, TKBMD00	TKBSTS01, TKBSTS00	Setting
Standalone mode (Period controlled by TKBCR00)	00B	00B	Not available
Standalone mode (Period controlled by external trigger input)	00B	01B/10B/11B	Available
Interleave PFC output mode	11B	–	Available

**Remark** Available when the period is controlled by external trigger input.

**7.5.4 PWM output function for IH control**

Timer KB20 has the timer output function for IH control.

- IH-PWM output restart by INTPx, comparator detection 0, 1
- IH-PWM output restart by matching 16-bit counter and value set in compare register (TKBCR00)
- Forced output stop (high impedance) by detecting valid edge of INTP0
- Changing output pin by PWCTKB register when operation is stopped

<R> Make either of the settings to use the IH-PWM output function (TKBIHE0 = 1).

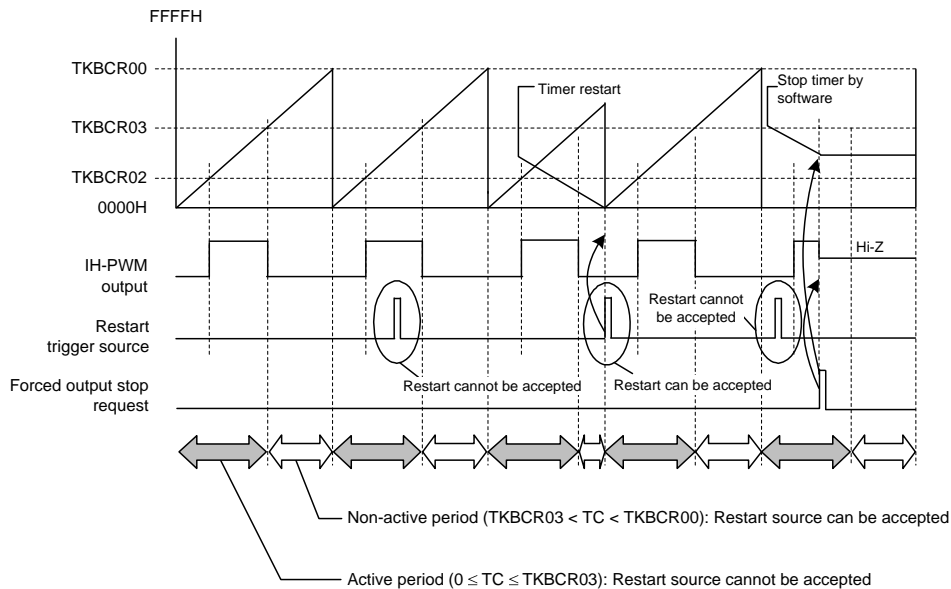
- Set a value other than 0000H in compare register 02 (TKBCR02).
- Set the timer clock select register (TKBPSCS0) and TKBCKS0 bit which selects the clock for timer KB20 so that the count clock ( $f_{KB20}$ ) for 16-bit timer KB20 is that running at  $f_{CLK}$  or  $f_{HOCO}$  with no division.

TKBO01-0, TKBO01-1, and TKBO01-2 are used to perform PWM output for IH control. The output of TKBO00 is the default level specified by using bit 0 (TKBTOD00) of 16-bit timer KB2 output control register 00 (TKBIOC00).

After the counter (TKBCNT0) starts, timer KB20 starts PWM output when the counter value reaches the count specified by the TKBCR02 register.

Timer KB20 continues high-level PWM output until the counter value reaches the value specified by the TKBCR03 register. The PWM output is active while the counter value is between 0000H and the value specified by the TKBCR03 register. During this period, no IH-PWM output restart requests can be acknowledged. The PWM output is inactive while the counter value is between the value specified by the TKBCR03 register and the value specified by the TKBCR00 register. During this period, IH-PWM output restart requests can be acknowledged.

**Figure 7-65. PWM Output Function for IH Control**



**(1) Combinations of available operation mode**

Operation Mode	TKBMD01, TKBMD00	ELSELRn2, ELSELRn1, ELSELRn0	Setting
Standalone mode (Period controlled by TKBCR00)	00B	000B	Available
Standalone mode (Period controlled by external trigger input)	00B	100B	Available
Interleave PFC output mode	11B	-	Not available

**Caution** When using the PWM output function for IH control, set the TKBSTS01 and TKBSTS00 bits to 00B.

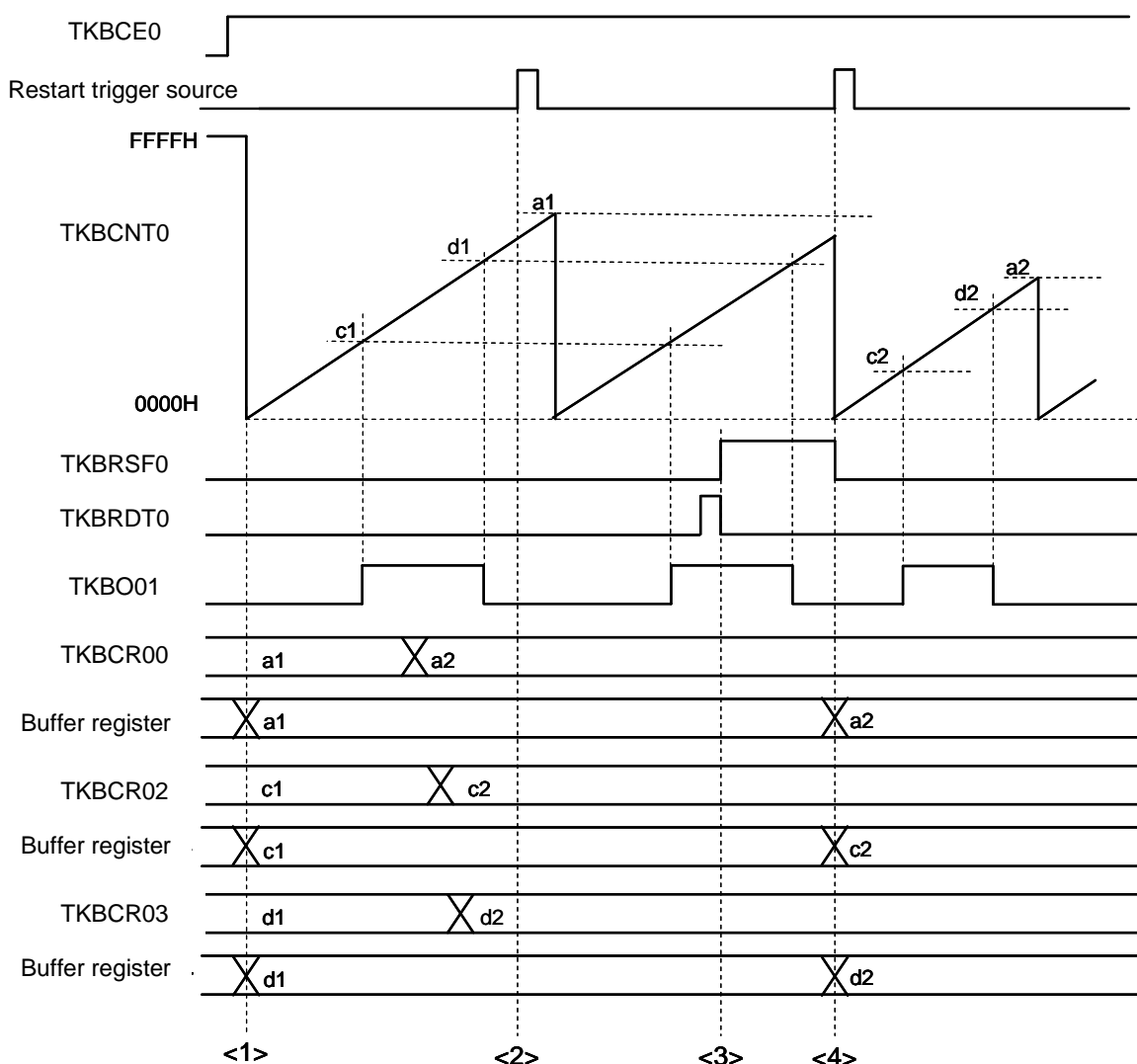
In standalone operation mode, batch overwrite can be controlled at the timing when restart trigger source is generated.

At the match of restart trigger source or TKBCNT0 and TKBCR00, counter clear and compare register batch overwrite are implemented within reception period after request by the TKBRDT0 bit and by setting the TKBTSE0 bit to 1.

Same as in counter clear, batch overwrite by match detection is also implemented when TKBCR00 and counter (TKBCNT0) match before generation of restart trigger source.

Even if the restart trigger source is generated, batch overwrite does not occur unless 1 is written to the TKBRDT0 bit.

**Figure 7-66. Updating Timing of Compare Registers 00, 02, 03 That Operates as PWM Output Function for IH Control**



- <1> Compare register setting is transferred to the buffer register at the timing when the TKBCE0 bit is set from 0 to 1 and TKBCNT0 starts counting operation.
- <2> After the TKBCR00, TKBCR02, and TKBCR03 registers are overwritten, even when restart trigger source is generated, batch overwrite is not generated if 1 is not written to the TKBRDT0 bit.
- <3> The batch overwrite trigger pending flag (TKBRSF0) is set to 1 by writing 1 to the TKBRDT0 bit.
- <4> Compare register setting is transferred to the buffer register by the restart trigger source generated when the TKBRSF0 bit is 1. The TKBRSF0 bit is set to 0 simultaneously.

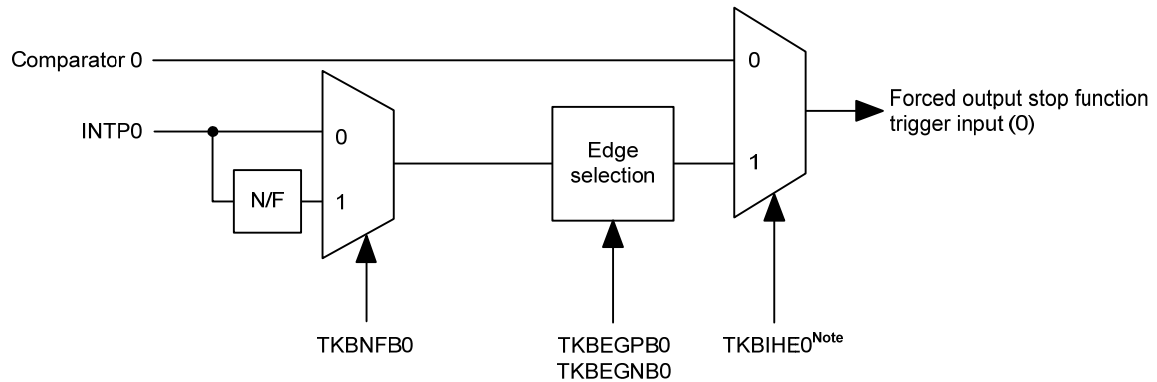
**(2) Trigger input selection for forced output stop function**

When the PWM output function for IH control is selected, the following can be selected.

- With/without noise filter
- Falling edge, rising edge, or both rising and falling edges can be selected

The following figure shows the circuit configuration.

**Figure 7-67. Circuit Configuration of Forced Output Stop Function Trigger Input Selection (0)  
When PWM Output Function for IH Control is Used**



**Note** When using the PWM output function for IH control, set  $TKBHE0 = 1$ .

### 7.6 Forced Output Stop Function

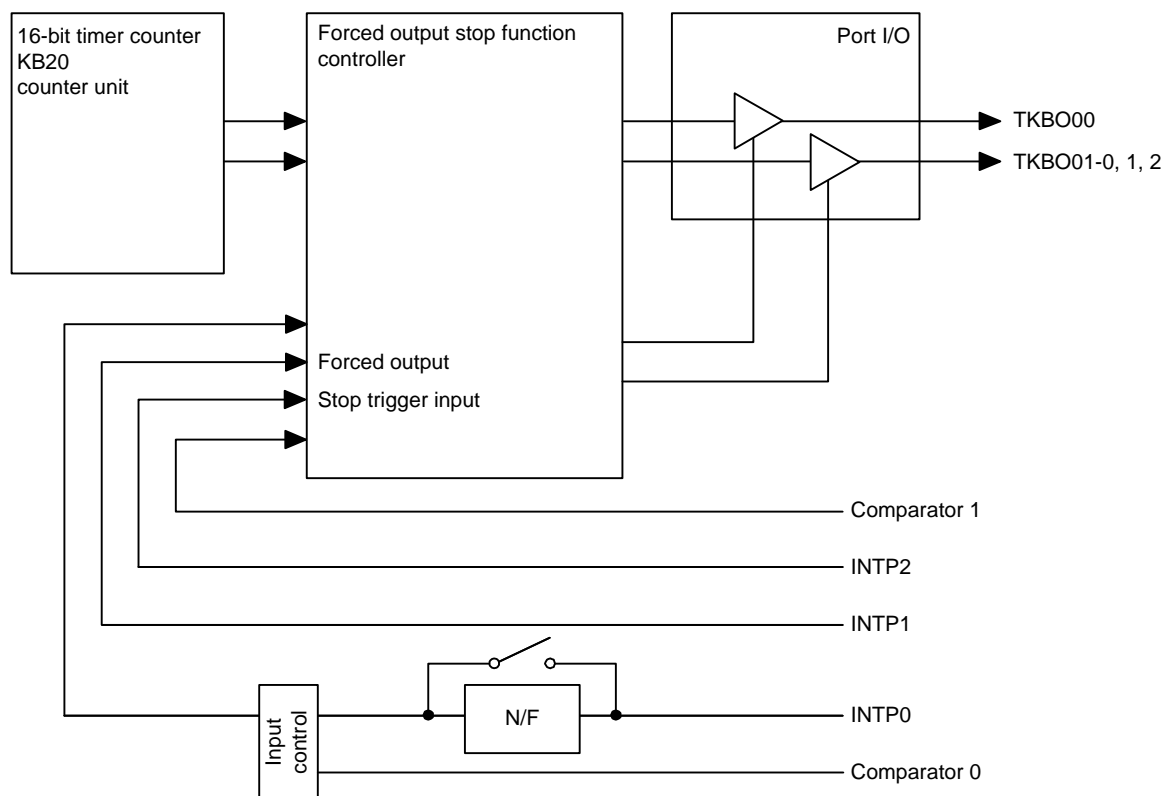
Forced output stop function is a function to protect power supply, etc.

If any abnormal situation that occurs in a power circuit configured outside of a microcomputer leads to over-voltage of over-current, making voltage or current sense signal into TIN0/comparator can protect the circuit by maintaining the timer output high impedance or fixed output state without being intermediated by a CPU's program control.

With this function, abnormality is identified only when input signal edge have been detected. Fixed level without edge is not recognized as an abnormality.

The following figure shows the system structure of forced output stop function.

**Figure 7-68. System Structure of Forced Output Stop Function**



### 7.6.1 Forced output stop functions 1 and 2

There are two control methods for the forced output stop function. Forced output stop function 1 can be set to fixed level output or high-impedance output, and forced output stop function 2 can set to fixed level output only.

The differences of the control methods are shown below.

#### (1) Selectable output levels for forced output stop functions 1 and 2

Selectable Output Levels	Forced Output Stop Function			
	When Using the PWM Output Function for IH Control (TKBIHE0 = 1)		When not Using the PWM Output Function for IH Control (TKBIHE0 = 0)	
	Function 1	Function 2	Function 1	Function 2
High-impedance output	√	–	√	–
Low-level fixed output	–	–	√	√
High-level fixed output	–	–	√	√

#### (2) Start/cancel conditions for forced output stop functions 1 and 2

Function/Operation Details (Start of Forced Output Stop)	Forced Output Stop Function			
	When Using the PWM Output Function for IH Control (TKBIHE0 = 1)		When not Using the PWM Output Function for IH Control (TKBIHE0 = 0)	
	Function 1	Function 2	Function 1	Function 2
Forced output stop starts when the valid edge of the comparator 1 or 0 output or the rising edge of INTP1 is detected	–	–	√	√
Forced output stop starts when the valid edge of INTP0 is detected	√	–	–	–
Forced output stop starts when the rising edge of INTP2 is detected	–	–	–	√
Forced output stop is stopped by software bit (TKBPAHTS0n) setting.	–	–	√	–

Function/Operation Details (Cancel of Forced Output Stop)	Forced Output Stop Function			
	When Using the PWM Output Function for IH Control (TKBIHE0 = 1)		When not Using the PWM Output Function for IH Control (TKBIHE0 = 0)	
	Function 1	Function 2	Function 1	Function 2
Forced output stop cancelled by software bit (TKBPAHTT0n) setting.	√	–	√	–
Forced output stop cancelled in synchronization with TMKB period after the software bit (TKBPAHTT0n) setting.	–	–	√	–
Forced output stop cancelled at the next TMKB period after the start of forced output stop.	–	–	–	√
Forced output stop cancelled in synchronization with TMKB period after the detection of the start edge and reverse edge.	–	–	–	√



**(3) Conditions of selectable input pins and available trigger bits for forced output stop functions 1 and 2**

Selectable Input Pins	Forced Output Stop Function			
	When Using the PWM Output Function for IH Control (TKBIHE0 = 1)		When not Using the PWM Output Function for IH Control (TKBIHE0 = 0)	
	Function 1	Function 2	Function 1	Function 2
External input (INTP0) <sup>Note</sup>	√	–	–	–
External input (INTP1)	–	–	√	√
External input (INTP2)	–	–	–	√
Comparator 0	–	–	√	√
Comparator 1	–	–	√	√

**Note** Whether the noise filter is used or not can be selected for INTP0.

Available Trigger Bits	Forced Output Stop Function			
	When Using the PWM Output Function for IH Control (TKBIHE0 = 1)		When not Using the PWM Output Function for IH Control (TKBIHE0 = 0)	
	Function 1	Function 2	Function 1	Function 2
Software bit (TKBPAHTS0n)	–	–	√	–
Software bit (TKBPAHTT0n)	√	–	√	–

## 7.7 Operation of Forced Output Stop Function 1

### 7.7.1 I/O setting for forced output stop function 1

In forced output stop function 1, comparator 0/INTP0, comparator 1, or INTP1 is used as the trigger signal of forced output stop. Selectable output status for forced output stop is high impedance or high/low-level fixed. The tables below show trigger signal selection and output status settings.

#### (1) TKBO00 output control

• Forced output stop function trigger selection

TKBPACTL00	Input Selection
TKBPAHVS002	External interrupt (INTP1)
TKBPAHVS001	Comparator 1
TKBPAHVS000	Comparator 0

• Output selection

TKBPACTL00		Output Status
TKBPAMD001	TKBPAMD000	
0	0	High-impedance output
0	1	High-impedance output
1	0	Low-level fixed output
1	1	High-level fixed output

• Start of forced output stop function 1

TKBPACTL00		Start Condition Selection for Forced Output Stop Function 1
TKBPAHCM001	TKBPAHCM000	
0	0	The forced output stop function starts operating when an input that triggers the forced output stop function is detected, or when 1 is written to the TKBPAHTS00 bit.
0	1	
1	0	
1	1	

• Cancel of forced output stop function 1

TKBPACTL00		Cancel Condition Selection for Output of Forced Output Stop Function 1
TKBPAHCM001	TKBPAHCM000	
0	0	Forced output stop function 1 is cancelled by setting the TKBPAHTT00 bit to 1, regardless of the input level of forced output function stop input 1.
0	1	After the input of forced output stop function input 1 is cancelled, forced output stop function 1 is cancelled by setting the TKBPAHTT00 bit to 1. Setting the TKBPAHTT00 bit to 1 is invalid during the active period of the input.
1	0	After the TKBPAHTT00 bit is set to 1, forced output stop function 1 is cancelled in synchronization with the next restart of the counter, regardless of the input level of forced output stop function input 1.
1	1	After the input of forced output stop function input 1 is cancelled, forced output stop function 1 is cancelled in synchronization with the next restart of the counter after the TKBPAHTT00 bit is set to 1. Writing 1 to the TKBPAHTT00 bit is invalid during the active period of the input.

- Cautions**
1. Forced output stop function 1 cannot be used when using the PWM output function for IH control.
  2. If the comparator 0 or 1 detection signal is selected as a trigger input for forced output stop function 1 while the C0EDG and C1EDG bits of the comparator filter control register (COMPFIR) are 1 (both-edge detection), forced output stop function 1 cannot be canceled by setting TKBPAHCM000 to 1. To cancel forced output stop function 1, be sure to set TKBPAHCM000 to 0.
  3. INTP1 and INTP2, which are used to trigger forced output stop functions 1 and 2, are not affected by the setting of the external interrupt rising edge enable register (EGP0) and external interrupt falling edge enable register (EGN0).

**(2) TKBO01-0, TKBO01-1, TKBO01-2 output control**

## • Forced output stop function trigger selection

TKBPACTL01	Input Selection	When Using the PWM Output Function for IH Control (TKBIHE0 = 1)	When not Using the PWM Output Function for IH Control (TKBIHE0 = 0)
TKBPAHVS012	External interrupt (INTP1)	–	√
TKBPAHVS011	Comparator 1	–	√
TKBPAHVS010	External interrupt (INTP0)	√	–
	Comparator 0	–	√

## • Output selection

TKBPACTL01		Output Status	When Using the PWM Output Function for IH Control (TKBIHE0 = 1)	When not Using the PWM Output Function for IH Control (TKBIHE0 = 0)
TKBPAMD011	TKBPAMD010			
0	0	High-impedance output	√	√
0	1	High-impedance output	–	√
1	0	Low-level fixed output	–	√
1	1	High-level fixed output	–	√

## • Start of forced output stop function 1

TKBPACTL01		Start Condition Selection for Forced Output Stop Function 1	When Using the PWM Output Function for IH Control (TKBIHE0 = 1)	When not Using the PWM Output Function for IH Control (TKBIHE0 = 0)
TKBPAMD011	TKBPAMD010			
0	0	The forced output stop function starts operating when an input that triggers the forced output stop function is detected, or when 1 is written to the TKBPAHVS01 bit.	√	√
0	1		–	√
1	0		–	√
1	1		–	√

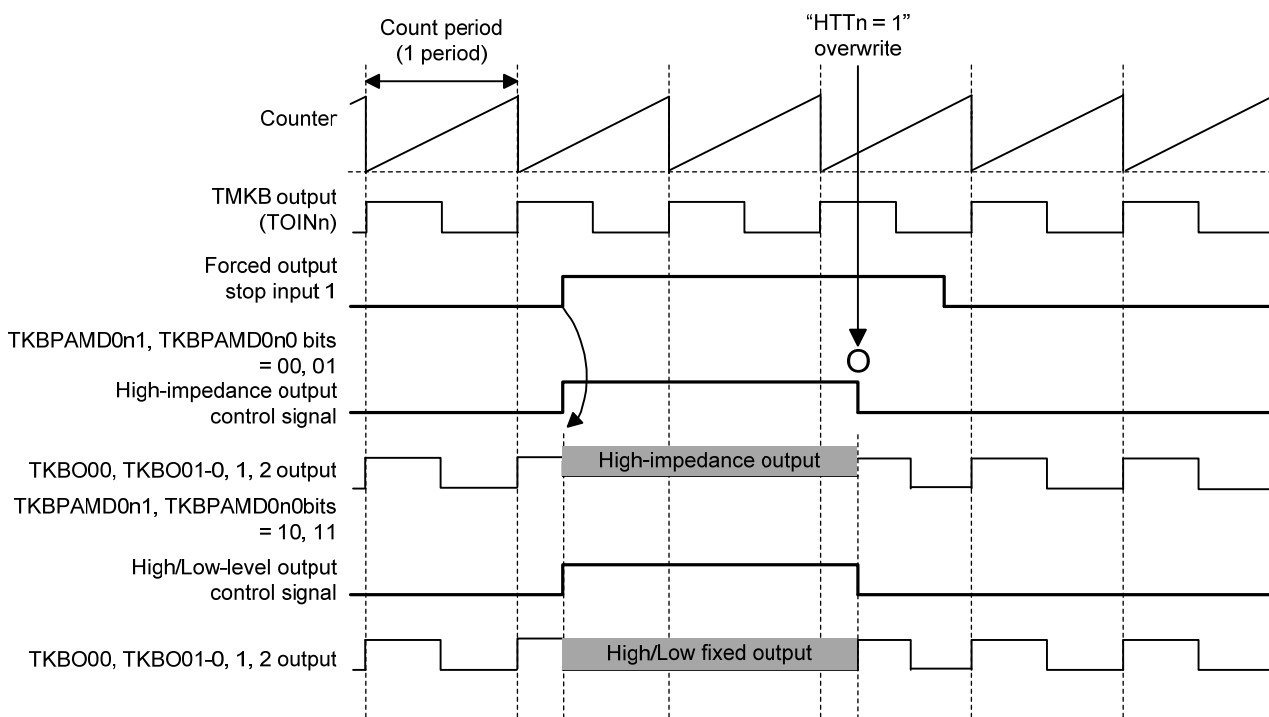
• Cancel of forced output stop function 1

TKBPACTL01		Cancel Condition Selection for Output of Forced Output Stop Function 1	When Using the PWM Output Function for IH Control (TKBIHE0 = 1)	When not Using the PWM Output Function for IH Control (TKBIHE0 = 0)
TKBPAMD011	TKBPAMD010			
0	0	Forced output stop function 1 is cancelled by setting the TKBPAHTT01 bit to 1, regardless of the input level of forced output function stop input 1.	√	√
0	1	After the input of forced output stop function input 1 is cancelled, forced output stop function 1 is cancelled by setting the TKBPAHTT01 bit to 1. Setting the TKBPAHTT01 bit to 1 is invalid during the active period of the input.	–	√
1	0	After the TKBPAHTT01 bit is set to 1, forced output stop function 1 is cancelled in synchronization with the next restart of the counter, regardless of the input level of forced output stop function input 1.	–	√
1	1	After the input of forced output stop function input 1 is cancelled, forced output stop function 1 is cancelled in synchronization with the next restart of the counter after the setting TKBPAHTT01 bit is set to 1. Writing 1 to the TKBPAHTT00 bit is invalid during the active period of the input.	–	√

- Cautions**
1. If the comparator 0 or 1 detection signal is selected as a trigger input for forced output stop function 1 while the C0EDG and C1EDG bits of the comparator filter control register (COMPFIR) are 1 (both-edge detection), forced output stop function 1 cannot be canceled by setting TKBPAHCM010 to 1. To cancel forced output stop function 1, be sure to set TKBPAHCM010 to 0.
  2. INTP1 and INTP2, which are used to trigger forced output stop functions 1 and 2, are not affected by the setting of the external interrupt rising edge enable register (EGP0) and external interrupt falling edge enable register (EGN0). Only the rising edge is valid. Use the TKBEGPB0 and TKBEGNB0 bits of 16-bit timer KB2 output control register 01 (TKBIOC01) to select the valid edge of the INTP0 signal used by forced output stop function 1.

7.7.2 Basic operation of forced output stop function 1

(1) TKBPAHCM0n1, TKBPAHCM0n0 bits = 00



**Remark** n = 0, 1

• TKBPAMD0n1, TKBPAMD0n0 bits = 00, 01

When the rising edge of forced output stop input 1 is detected, the high-impedance output control signal is set to high level and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output becomes high impedance.

Regardless of the level of forced output stop input 1, the high-impedance output control signal is set to low level by writing 1 to the TKBPAHTT0n bit in the TKBPAHFT0 register, and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 return to PWM output.

The high-level period of the high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

• TKBPAMD0n1, TKBPAMD0n0 bits = 10, 11

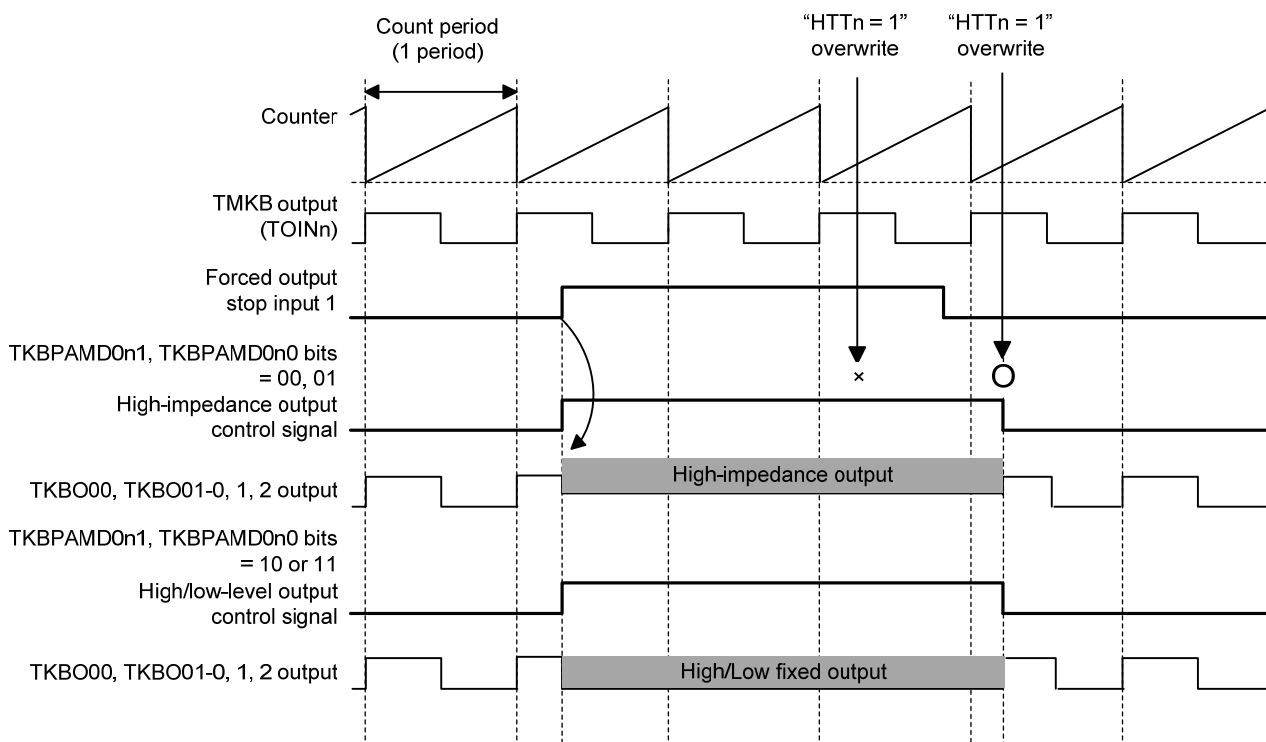
When the rising edge of forced output stop input 1 is detected, the high/low-level output control signal is set to high level and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output is fixed to high/low level according to the setting of the TKBPAMD0n bit.

Regardless of the level of forced output stop input 1, the high/low-level output control signal is set to low level by writing 1 to the TKBPAHTT0n bit in the TKBPAHFT0 register, and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 return to PWM output.

The high-level period of the high/low-level output control signal is the period for forced output stop 1 (high/low-level fixed output).

**Caution** When using the PWM output function for IH control, set the TKBPAHCM0n1 and TKBPAHCM0n0 bits to 00, and the TKBPAMD0n1 and TKBPAMD0n0 bits to 00.

(2) TKBPAHCM0n1, TKBPAHCM0n0 bits = 01



**Remark** n = 0, 1

• TKBPAMD0n1, TKBPAMD0n0 bits = 00, 01

When the rising edge of forced output stop input 1 is detected, the high-impedance output control signal is set to high level and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output becomes high impedance.

Writing 1 to the TKBPAHTT0n bit in the TKBPAHFT0 register is invalid during the high-level period of the input of forced output stop input 1. After forced output stop input 1 changes to low level, the high-impedance output control signal is set to low level by writing 1 to the TKBPAHTT0n bit, and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 return to PWM output.

The high-level period of the high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

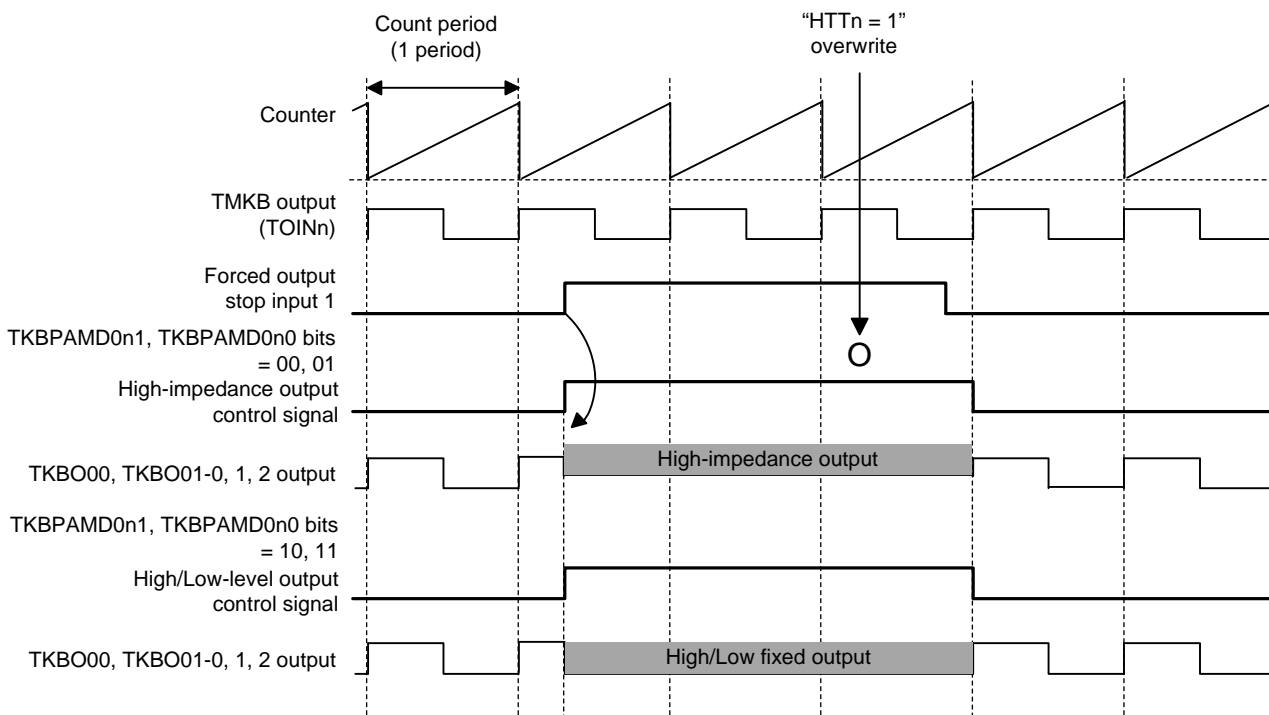
• TKBPAMD0n1, TKBPAMD0n0 bits = 10, 11

When the rising edge of forced output stop input 1 is detected, the high/low-level output control signal is set to high level and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output is fixed to high/low level according to the setting of the TKBPAMD00n bit.

Writing 1 to the TKBPAHTT0n bit in the TKBPAHFT0 register is invalid during the high-level period of the input of forced output stop input 1. After input of forced output stop input 1 changes to low level, the fixed low-level/high-level of TKBPATFOUTn is cancelled by writing 1 to the TKBPAHTT0n bit, and PWM is output from TKBO00, TKBO01-0, TKBO01-1, TKBO01-2.

**Caution** When using the PWM output function for IH control, the TKBPAHCM0n1 and TKBPAHCM0n0 bits cannot be set to 01.

**(3) TKBPAHCM0n1, TKBPAHCM0n0 bits = 10**



**Remark** n = 0, 1

• TKBPAMD0n1, TKBPAMD0n0 bits = 00, 01

When the rising edge of forced output stop input 1 is detected, the high-impedance output control signal is set to high level and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output becomes high impedance.

Regardless of the level of forced output stop input 1, the high-impedance output control signal is set to low level in synchronization with the restart of the TMKB counter by writing 1 to the TKBPAHTT0n bit.

The high-level period of the high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

• TKBPAMD0n1, TKBPAMD0n0 bits = 10, 11

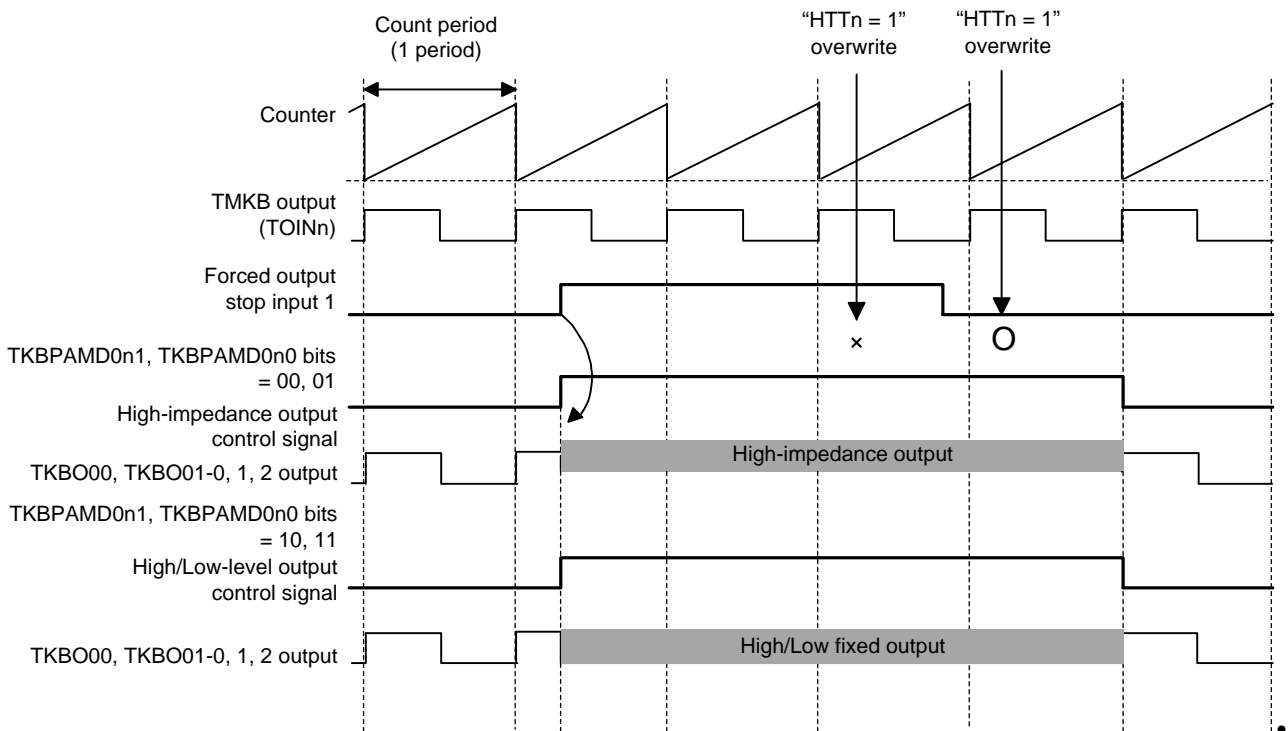
When the rising edge of forced output stop input 1 is detected, the high/low-level output control signal is set to high level and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output is fixed to high/low level according to the setting of the TKBPAMD00n bit.

Regardless of the level of forced output stop input 1, the high/low-level output control signal is set to low level in synchronization with the restart of the TMKB counter by writing 1 to the TKBPAHTT0n bit, and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 return to PWM output.

The high-level period of the high/low-level output control signal is the period for forced output stop 1 (high/low-level fixed output).

**Caution** When using the PWM output function for IH control, the TKBPAHCM0n1 and TKBPAHCM0n0 bits cannot be set to 10.

(4) TKBPAHCM0n1, TKBPAHCM0n0 bits = 11



**Remark** n = 0, 1

• TKBPAMD0n1, TKBPAMD0n0 bits = 00, 01

When the rising edge of forced output stop input 1 is detected, the high-impedance output control signal is set to high level and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output becomes high impedance.

Writing 1 to the TKBPAHTT0n bit in the TKBPAHFT0 register is invalid during the high-level period of the input of forced output stop input 1.

After forced output stop input 1 changes to low level, the high-impedance output control signal is set to low level in synchronization with the restart of the TMKB counter by writing 1 to the TKBPAHTT0n bit.

The high-level period of the high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

• TKBPAMD0n1, TKBPAMD0n0 bits = 10, 11

When the rising edge of forced output stop input 1 is detected, the high/low-level output control signal is set to high level and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 output is fixed to high/low level according to the setting of the TKBPAMD00n bit.

The high/low-level output control signal is set to low level in synchronization with the restart of the TMKB counter by writing 1 to the TKBPAHTT0n bit in the TKBPAHFT0 register during the high-level period of forced output stop 1, and TKBO00, TKBO01-0, TKBO01-1, TKBO01-2 return to PWM output.

The high-level period of the high/low-level output control signal is the period for forced output stop 1 (high/low-level fixed output).

**Caution** When using the PWM output function for IH control, the TKBPAHCM0n1 and TKBPAHCM0n0 bits cannot be set to 11.



### 7.7.3 Notes on using forced output stop function 1

1. When  $TKBPAHCMn = 10$  or  $11$ , forced output stop is cancelled as follows.

#### (1) Setting $TKBPAHCM0n = 10$

##### (a) When forced output stop input 1 occurs

When  $TKBPAHCM0n = 10$ , if forced output stop input 1 is detected after setting  $TKBPAHTT0n = 1$  and before the restart of the counter, stop input is ignored, and forced output stop is cancelled at the next restart of the counter.

##### (b) When $TKBPAHTS0n$ is set to 1

When  $TKBPAHCM0n = 10$ , if  $TKBPAHTS0n$  is set to 1 after setting  $TKBPAHTT0n = 1$  and before the next restart of the counter, setting  $TKBPATHTT0n = 1$  is invalid, and forced output stop is not cancelled at the next restart of the counter. To cancel forced output stop, set  $TKBPAHTT0n$  to 1 again.

#### (2) Setting $TKBPAHCM0n = 11$

##### (a) Forced output stop input 1 occurs

When  $TKBPAHCM0n = 11$ , if forced output stop input 1 is detected after setting  $TKBPAHTT0n = 1$  and before the next counter period, setting  $TKBPATHTT0n = 1$  is invalid, and forced output stop is not cancelled at the next restart of the counter. To cancel forced output stop, set  $TKBPAHTS0n$  to 1 again.

##### (b) $TKBPAHTS0n$ is set to 1

When  $TKBPAHCM0n = 11$ , if  $TKBPAHTS0n$  is set to 1 after setting  $TKBPAHTT0n = 1$  and before the next counter period, setting  $TKBPATHTT0n = 1$  is invalid and, forced output stop is not cancelled at the next restart of the counter. To cancel forced output stop, set  $TKBPAHTT0n$  to 1 again.

2. Timing for setting  $TKBPAHTS0n$  and  $TKBPATHTT0n$  when  $TKBPAHCM0n = 01, 11$

When  $TKBPAHCM0n = 01, 11$ , wait until 1  $f_{CLK}$  clock elapses before setting  $TKBPATHTT0n$  to 1 after setting  $TKBPAHTS0n = 1$ .

## 7.8 Operation of Forced Output Stop Function 2

### 7.8.1 I/O setting for forced output stop function 2

For forced output stop function 2, comparator 0, comparator 1, INTP1, or INTP2 is used as the trigger signal to stop forced output.

The output status for forced output stop is fixed to high/low-level. The tables below show trigger signal selection and output status settings.

#### (1) TKBO00 output control

• Forced output stop function trigger selection

TKBPACTL00	Input Selection
TKBPAFXS003	External interrupt (INTP2)
TKBPAFXS002	External interrupt (INTP1)
TKBPAFXS001	Comparator 1
TKBPAFXS000	Comparator 0

• Output selection

TKBPACTL00		Output Status
TKBPAMD001	TKBPAMD000	
0	0	Low-level fixed output
0	1	High-level fixed output
1	0	Low-level fixed output
1	1	High-level fixed output

**Caution** Operation of forced output stop function 2 does not affect the high-impedance output control signal. Do not select the high-impedance output control signal using the TKBPACTL00 register.

• Start of forced output stop function 2

TKBPACTL00	Start Condition Selection for Forced Output Stop Function 2
TKBPAFCM00	
0	Forced output stop function is started when forced output stop function trigger input is detected.
1	

• Cancel of forced output stop function 2

TKBPACTL00	Cancel Condition Selection for Output of Forced Output Stop Function 2
TKBPAFCM00	
0	Forced output stop function 2 is started, and cancelled in synchronization with the next restart of the counter.
1	

- Cautions**
1. If the comparator 0 or 1 detection signal is selected as a trigger input for forced output stop function 2 while the C0EDG and C1EDG bits of the comparator filter control register (COMPFIR) are 1 (both-edge detection), forced output stop function 2 cannot be canceled by setting TKBPAFCM00 to 1. To cancel forced output stop function 2, be sure to set TKBPAFCM00 to 0.
  2. INTP1 and INTP2, which are used to trigger forced output stop functions 1 and 2, are not affected by the setting of the external interrupt rising edge enable register (EGP0) and external interrupt falling edge enable register (EGN0).

**(2) TKBO01-0, TKBO01-1, TKBO01-2 output control**

• Forced output stop function trigger selection

TKBPACTL01	Input Selection
TKBPAFXS013	External interrupt (INTP2)
TKBPAFXS012	External interrupt (INTP1)
TKBPAFXS011	Comparator 1
TKBPAFXS010	Comparator 0

• Output selection

TKBPACTL01		Output Status
TKBPAMD011	TKBPAMD010	
0	0	Low-level fixed output
0	1	High-level fixed output
1	0	Low-level fixed output
1	1	High-level fixed output

**Caution** Operation of forced output stop function 2 does not affect the high-impedance output control signal. Do not select the high-impedance output control signal using the TKBPACTL01 register.

• Start of forced output stop function 2

TKBPACTL01	Start Condition Selection for Forced Output Stop Function 2
TKBPAFCM01	
0	Forced output stop function 2 is started when forced output stop function trigger input is detected.
1	

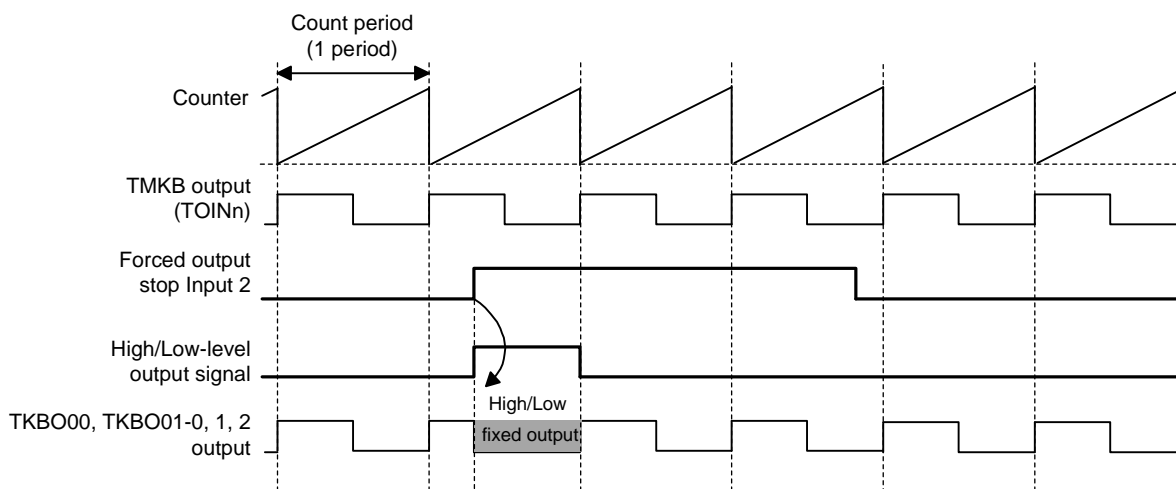
• Cancel of forced output stop function 2

TKBPACTL01	Output Cancel Condition Selection for Forced Output Stop Function 2
TKBPAFCM01	
0	Forced output stop function 2 is started, and cancelled in synchronization with the next restart of the counter.
1	Forced output stop function 2 is started, and cancelled in synchronization with the next restart of the counter after cancellation of the trigger is detected.

- Cautions**
1. If the comparator 0 or 1 detection signal is selected as a trigger input for forced output stop function 2 while the C0EDG and C1EDG bits of the comparator filter control register (COMPFIR) are 1 (both-edge detection), forced output stop function 2 cannot be canceled by setting TKBPAFCM01 to 1. To cancel forced output stop function 2, be sure to set TKBPAFCM01 to 0.
  2. INTP1 and INTP2, which are used to trigger forced output stop functions 1 and 2, are not affected by the setting of the external interrupt rising edge enable register (EGP0) and external interrupt falling edge enable register (EGN0). Only the rising edge is valid.

7.8.2 Basic operation of forced output stop function 2

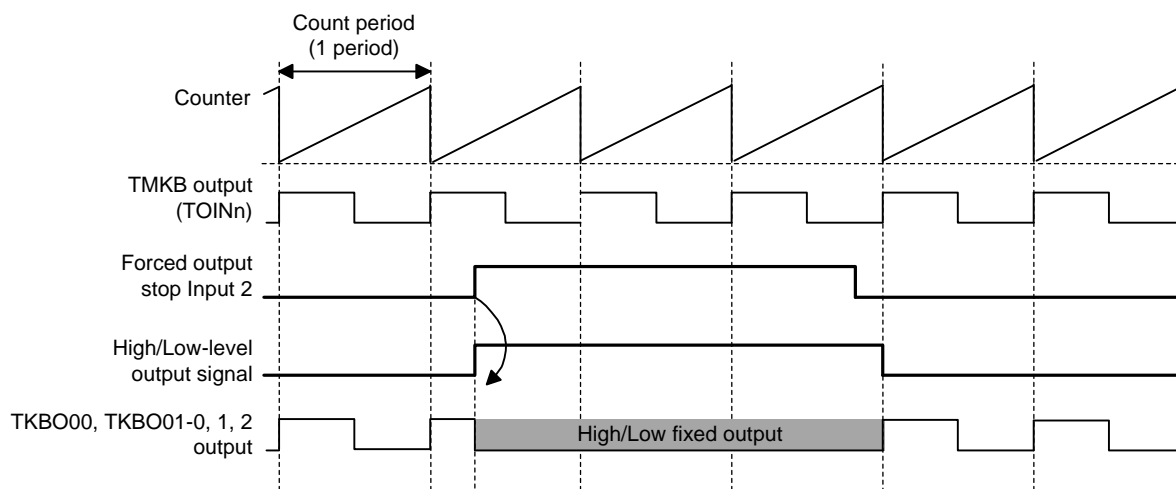
(1) Forced output stop function 2 when TKBPAFCM0n = 0



When the rising edge of forced output stop input 2 is detected, TKBPATFOUTn output is fixed to low or high level according to TKBPAMD00n setting.

Regardless of the input level of forced output stop input 2, the fixed level of TKBPATFOUTn is cancelled in synchronization with the next restart of the TMKB counter, and PWM is output from TKBO00, TKBO01-0, TKBO01-1, and TKBO01-2.

(2) Forced output stop function 2 when TKBPAFCM0n = 1



When the rising edge of forced output stop input 2 is detected, TKBPATFOUTn output is fixed to low or high level according to TKBPAMD00n setting.

After the rising edge of forced output stop input 2 is detected, the fixed level of TKBPATFOUTn is cancelled in synchronization with the next restart of the TMKB counter, and PWM is output from TKBO00, TKBO01-0, TKBO01-1, and TBO01-2.

## 7.9 Cautions When Using 16-bit Timer KB20

### 7.9.1 Using 16-bit timer KB20 together with the LIN-bus functions

16-bit timer KB20 cannot be used together with the LIN-bus functions. When using the LIN-bus functions (that is, when the input switch control register (ISC) is set to any value other than its initial value of 00H), be sure to set bit 4 (TKB2EN) of the peripheral enable register (PER1) to 0 (which sets timer KB20 to the reset status).

### 7.9.2 Cautions when using the counter restart trigger

#### (1) Using external interrupts (INTP0 to INTP7)

INTP0 to INTP7, which are used as counter restart trigger sources, are not affected by the settings of the external interrupt rising edge enable register (EGP0) or external interrupt falling edge enable register (EGN0). Only the rising edge is valid.

#### (2) Using comparator 0 and comparator 1 detection

When using comparator 0 and comparator 1 detection for counter restart trigger sources 0 to 2, the edge to be detected can be selected by setting bits 2, 3, 6, and 7 (C0EPO, C0EDG, C1EPO, and C1EDG) of the comparator filter control register (COMPFIR).

### 7.9.3 Cautions when using the forced output stop function (when not using the PWM output function for IH control)

#### (1) Using external interrupts (INTP1 and INTP2)

INTP1 and INTP2, which are used for forced output stop functions 1 and 2, are not affected by the settings of the external interrupt rising edge enable register (EGP0) or external interrupt falling edge enable register (EGN0). Only the rising edge is valid.

#### (2) Using comparator 0 and comparator 1 detection

When using comparator 0 and comparator 1 detection for forced output stop functions 1 and 2, the edge to be detected can be selected by setting bits 2, 3, 6, and 7 (C0EPO, C0EDG, C1EPO, and C1EDG) of the comparator filter control register (COMPFIR).

Note, however, that if bits 3 and 7 (C0EDG, and C1EDG) of the COMPFIR register are set to 1 (both-edge detection), TKBPAHCM0n0 = 1 cannot be selected as a condition for canceling forced output stop function 1. In this case, be sure to select TKBPAHCM0n0 = 0. Also, TKBPAPFM0n = 1 cannot be selected as a condition for canceling forced output stop function 2. Always select TKBPAPFM0n = 0.

**Remark** n = 0, 1

#### 7.9.4 Cautions when using the PWM output function for IH control

##### (1) Using other operation mode and function with PWM output function for IH control

Interleave PFC output mode, PWM output dithering function, PWM output smooth start function, and maximum frequency limit function cannot be used together with the PWM output function for IH control.

##### (2) Selecting the counter restart trigger source

When using the PWM output function for IH control, counter restart trigger sources 0, 1, and 2 cannot be selected. Be sure to set bits 1 and 0 of 16-bit timer KB2 operation control register 00 (TKBCTL00) to their initial value (00).

##### (3) Timer output

Timer output (TKBO00) cannot be used when using the PWM output function for IH control. Be sure to set bit 0 of 16-bit timer KB2 output control register 01 (TKBIOC01) to its initial value (0).

Only combination of low/high or high/low can be used for setting of default level/active level. Be sure to set bits 3 and 1 of 16-bit timer KB2 output control register 0 (TKBIOC00) to "00" or "11".

##### (4) Cautions when using the forced output stop function

When using the forced output stop function, the PWM output function for IH control can be used under the following conditions:

- Only the control settings related to timer output (TKBO01-0, TKBO01-1, and TKBO01-2) can be specified.
- Only forced output stop function 1 can be used.
- When the forced output stop function is executed, only high-impedance output can be selected.
- Only INTP0 can be selected as the trigger for starting the forced output stop function.
- Software cannot be used to start the forced output stop function.

Therefore, use the registers that control the forced output stop function under the following conditions:

Be sure to set forced output stop function control register 0 (TKBPACTL00) to its initial value (0000H).

In forced output stop function control register 1 (TKBPACTL01), only bit 4 (TKBPAHVS010) can be used. Be sure to set the other bits in this register to their initial value (0).

In forced output stop function control register 2 (TKBPACTL02), only bit 1 (TKBPACE01) can be used. Be sure to set bit 0 (TKBPACE00) to its initial value (0).

In the forced output stop function flag register (TKBP AFLG0), only bit 6 (TKBPAHSF01) can be used.

Do not use the other bits.

The forced output stop function 1 start register (TKBPAHFS0) cannot be used. Be sure to set this register to its initial value (00H).

In the forced output stop function 1 stop register (TKBPAHFT0), only bit 1 (TKBPAHTT01) can be used. Be sure to set bit 0 (TKBPAHTT00) to its initial value (0).

**CHAPTER 8 REAL-TIME CLOCK 2**

**8.1 Functions of Real-time Clock 2**

Real-time clock 2 (RTC2) has the following functions.

- Counters of year, month, day of the week, date, hour, minute, and second, that can count up to 99 years (with leap year correction function)
- Constant-period interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm interrupt function (alarm: day of the week, hour, and minute)
- Pin output function of 1 Hz

The real-time clock 2 interrupt signal (INTRTC) can be utilized for wakeup from STOP mode and triggering an A/D converter’s SNOOZE mode.

**Caution** The year, month, week, day, hour, minute and second can only be counted when a subsystem clock ( $f_{SUB} = 32.768 \text{ kHz}$ ) is selected as the operation clock of real-time clock 2. When the low-speed oscillation clock ( $f_{IL} = 15 \text{ kHz}$ ) is selected, only the constant-period interrupt function is available. However, the constant-period interrupt interval when  $f_{IL}$  is selected will be calculated with the constant-period (the value selected with RTCC0 register)  $\times f_{SUB}/f_{IL}$ .

**8.2 Configuration of Real-time Clock 2**

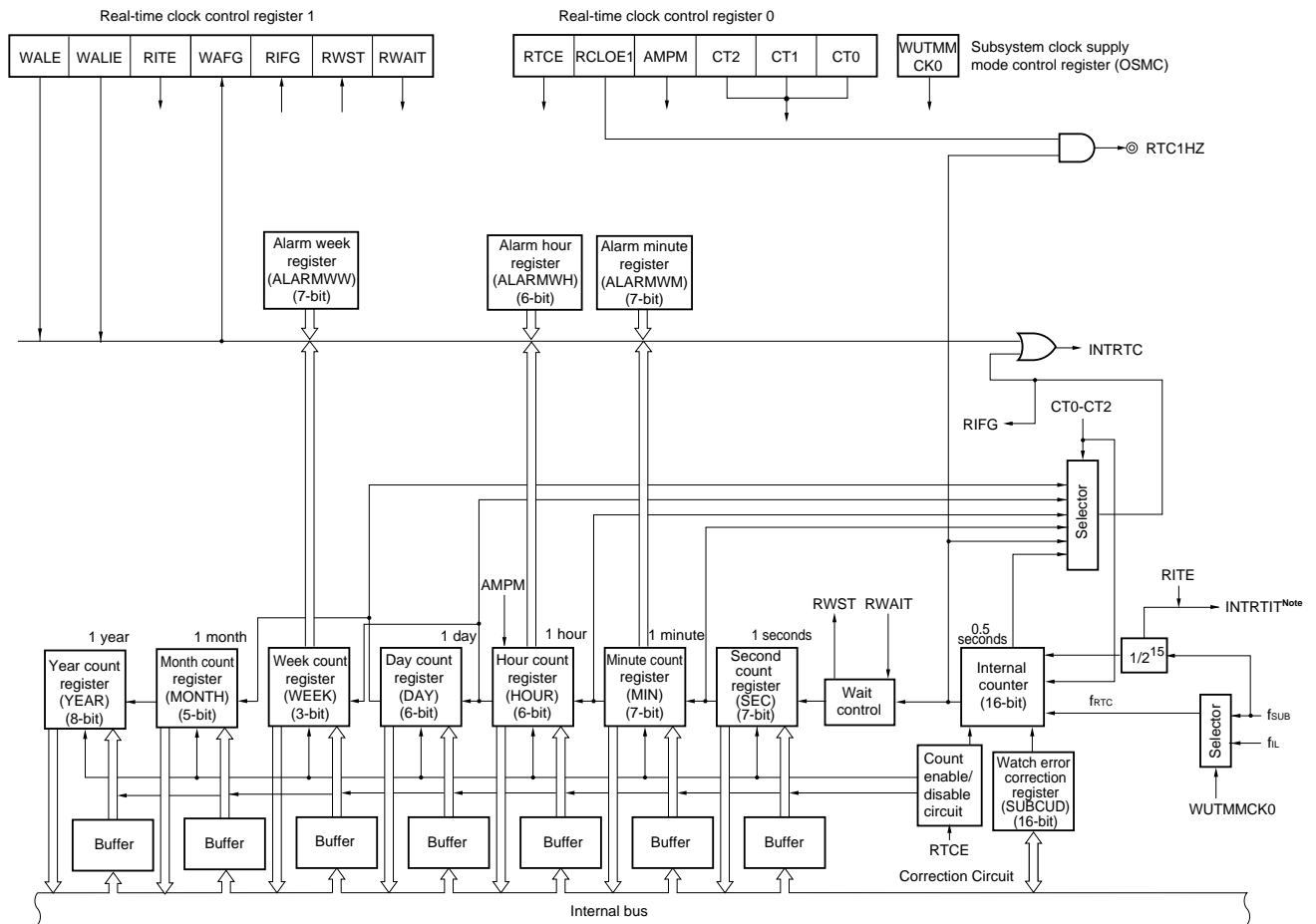
Real-time clock 2 includes the following hardware.

**Table 8-1. Configuration of Real-time Clock 2**

Item	Configuration
Counter	Counter (16-bit)
Control registers	Peripheral enable register 0 (PER0)
	Subsystem clock supply mode control register (OSMC)
	Power-on-reset status register (PORSR)
	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 8-1 shows the real-time clock 2 diagram.

Figure 8-1. Real-time Clock 2 Diagram



**Note** An interrupt that indicates the timing to get the correction value from the clock error correction register (SUBCUD). The fetch timing is 1 second ( $f_{SUB}$  base) interval.



### 8.3 Registers Controlling Real-time Clock 2

Real-time clock 2 is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- Power-on-reset status register(PORSR)
- 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)

The following shows the register states depending on reset sources.

Reset Source	System Registers <sup>Note 1</sup>	Calendar Registers <sup>Note 2</sup>
POR	Reset	Not reset
External reset	Retained	Retained
WDT	Retained	Retained
TRAP	Retained	Retained
LVD	Retained	Retained
Other internal reset	Retained	Retained

**Notes 1.** RTCC0, RTCC1, and SUBCUD

**2.** SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, ALARMWW, (counter)

Reset generation does not reset the SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, or ALARMWW registers. Initialize all the registers after power on.

### 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 real-time clock 2 registers are manipulated, be sure to set bit 7 (RTCWEN) 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	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

RTCWEN	Control of internal clock supply to real-time clock 2
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by real-time clock 2 cannot be written.</li> <li>• Real-time clock 2 can operate.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>• SFR used by real-time clock 2 can be read/written.</li> <li>• Real-time clock 2 can operate.</li> </ul>

- Cautions**
1. The clock error correction register (SUBCUD) becomes read/write enabled when RTCWEN in the peripheral enable register 0 (PER0) is set to 1.
  2. When using real-time clock 2, first set the RTCWEN bit to 1 and then set the following registers, while oscillation of the count clock (f<sub>RTC</sub>) is stable. If RTCWEN = 0, writing to the control registers of real-time clock 2 is ignored, and read values are the values set when RTCWEN = 1 (except for the subsystem clock supply mode control register (OSMC), power-on reset status register (PORSR), 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)
  3. Be sure to set bits 6 and 1 to 0.

### 8.3.2 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 other than real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver is stopped in STOP mode or in HALT mode while the subsystem clock is selected as the CPU clock.

In addition, the OSMC register is used to select the operation clock of real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver.

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 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	In STOP mode and in HALT mode while the CPU operates using the subsystem clock
0	Enables subsystem clock supply to peripheral functions. For peripheral functions for which operation is enabled, see <b>Tables 21-1 to 21-3</b> .
1	Stops subsystem clock supply to peripheral functions other than real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver.

WUTMMCK0	Selection of operation clock of real-time clock 2, 12-bit interval timer, and LCD controller/driver	Selection of clock output from PCLBUZn pin of clock output/buzzer output controller
0	Subsystem clock ( $f_{SUB}$ )	Selecting the subsystem clock ( $f_{SUB}$ ) is enabled.
1	Low-speed on-chip oscillator clock ( $f_{IL}$ )	Selecting the subsystem clock ( $f_{SUB}$ ) is disabled.

- Cautions 1. If the subsystem clock is oscillating, be sure to select the subsystem clock (WUTMMCK0 bit = 0).**
- 2. When WUTMMCK0 is set to 1, the low-speed on-chip oscillator clock oscillates.**
  - 3. When WUTMMCK0 is set to 1, only the constant-period interrupt function of real-time clock 2 can be used. The year, month, day of the week, day, hour, minute, and second counters and the 1 Hz output function of real-time clock 2 cannot be used. The interval of the constant-period interrupt is calculated by constant period (value selected by using the RTCC0 register)  $\times f_{SUB}/f_{IL}$ .**
  - 4. The subsystem clock and low-speed on-chip oscillator clock can only be switched by using the WUTMMCK0 bit if real-time clock 2, 12-bit interval timer, and LCD controller/driver are all stopped.**

**8.3.3 Power-on-reset status register (PORSR)**

The PORSR register is used to check the occurrence of a power-on reset.  
 Writing 1 to bit 0 (PORF) of the PORSR register enables this function. Writing 0 disables this function.  
 Write 1 to the PORF bit in advance to enable checking of the occurrence of a power-on reset.  
 The PORSR register can be set by an 8-bit memory manipulation instruction.  
 Power-on reset signal generation clears this register to 00H.

- Cautions 1. The PORSR register is reset only by a power-on reset; it retains the value when a reset caused by another factor occurs.**
- 2. If the PORF bit is set to 1, it guarantees that no power-on reset has occurred, but it does not guarantee that the RAM value is retained.**

**Figure 8-4. Format of Power-on-Reset Status Register (PORSR)**

Address: F00F9H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
PORSR	0	0	0	0	0	0	0	PORF

PORF	Checking occurrence of power-on reset
0	A value 1 has not been written, or a power-on reset has occurred.
1	No power-on reset has occurred.

**8.3.4 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 2 operation, control the RTC1HZ pin, set the 12- or 24-hour system, and set the constant-period interrupt function.

RTCC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Internal reset generated by the power-on-reset circuit clears this register to 00H.

**Figure 8-5. Format of Real-time Clock Control Register 0 (RTCC0) (1/2)**

Address: FFF9DH After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1	0	AMPM	CT2	CT1	CT0

RTCE <sup>Note 1</sup>	Real-time clock 2 operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1 <sup>Note 2</sup>	RTC1HZ pin output control
0	Disables output of the RTC1HZ pin (1 Hz)
1	Enables output of the RTC1HZ pin (1 Hz)
Output of 1 Hz is not output because the clock counter does not operate when RTCE = 0.	

- Notes**
1. When shifting to STOP mode immediately after setting RTCE to 1, use the procedure shown in **Figure 8-19 Procedure for Shifting to HALT/STOP Mode After Setting RTCE = 1**.
  2. When the RCLOE1 bit is set while the clock counter operates (RTCE = 1), a glitch may be output to the 1 Hz output pin (RTC1HZ).

**Caution** Be sure to clear bits 4 and 6 to “0”.

**Figure 8-5. Format of Real-time Clock Control Register 0 (RTCC0) (2/2)**

Address: FFF9DH After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1	0	AMPM	CT2	CT1	CT0

**Table 8-2. Relationship Between RTCE and RCLOE1 Settings and Status**

Register Settings		Status	
RTCE	RCLOE1	Real-time clock 2	RTC1HZ pin output
0	×	Counting stopped	No output
1	0	Count operation	No output
	1	Count operation	1 Hz output

AMPM	12-/24-hour system select
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system

When changing the value of the AMPM bit while the clock counter operates (RTCE = 1), set RWAIT (bit 0 of RTCC1) and then set the hour counter (HOUR) again.  
 When the AMPM value is 0, the 12-hour system is displayed. When the value is 1, the 24-hour system is displayed. Table 8-3 shows the displayed time digits.

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.

**Caution** Be sure to clear bits 4 and 6 to “0”.

**Remark** ×: don't care

**8.3.5 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.

RTCC1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Internal reset generated by the power-on-reset circuit clears this register to 00H.

**Figure 8-6. Format of Real-time Clock Control Register 1 (RTCC1) (1/3)**

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	RITE	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.

**Caution** If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8-bit manipulation instruction.

To prevent the RIFG and WAFG flags from being cleared during writing, set the corresponding bit to 1 (to disable writing). If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.

**Figure 8-6. Format of Real-time Clock Control Register 1 (RTCC1) (2/3)**

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	RITE	WAFG	RIFG	0	RWST	RWAIT

RITE	Control of correction timing signal interrupt (INTRTIT) function operation
0	Does not generate interrupt of correction timing signal.
1	Generates interrupt of correction timing signal.

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 clock (32.768 kHz) after matching of the alarm is detected. This flag is cleared when "0" is written to it. Writing "1" to it is invalid.	

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.	

**Caution** If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8-bit manipulation instruction.

To prevent the RIFG and WAFG flags from being cleared during writing, set the corresponding bit to 1 (to disable writing). If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.



**Figure 8-6. Format of Real-time Clock Control Register 1 (RTCC1) (3/3)**

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	RITE	WAFG	RIFG	0	RWST	RWAIT

RWST	Wait status flag of real-time clock 2
0	Counter is operating.
1	Mode to read or write counter value.
<p>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.                  Even if the RWAIT bit is set to 0, the RWST bit is not set to 0 while writing to the counter. After writing is completed, the RWST bit is set to 0.</p>	

RWAIT	Wait control of real-time clock 2
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value.
<p>This bit controls the operation of the counter.                  Be sure to write "1" to it to read or write the counter value.                  As the counter (16-bit) is continuing to run, complete reading or writing within one second and turn back to 0.                  When RWAIT = 1, it takes up to one cycle of <math>f_{RTC}</math> until the counter value can be read or written (RWST = 1).<sup>Note1, 2</sup>                  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.</p>	

**Note1.** 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".

**Note2.** 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 RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8-bit manipulation instruction.

To prevent the RIFG and WAFG flags from being cleared during writing, set the corresponding bit to 1 (to disable writing). If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.

**Remarks 1.** Constant-period 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 constant-period interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

**2.** The internal counter (16 bits) is cleared when the second count register (SEC) is written.

### 8.3.6 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 is a decimal counter that counts up when the counter (16-bit) overflows. When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks of `fRTC` 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 does not clear this register to its default value.

**Figure 8-7. Format of Second Count Register (SEC)**

Address: FFF92H    After reset: Undefined    R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

**Remark** The internal counter (16 bits) is cleared when the second count register (SEC) is written.

**Caution** When reading or writing to SEC while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 counter and 8.4.4 Writing to real-time clock 2 counter.

### 8.3.7 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 is a decimal counter that 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 2 clocks of `fRTC` 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 does not clear this register to its default value.

**Figure 8-8. Format of Minute Count Register (MIN)**

Address: FFF93H    After reset: Undefined    R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

**Caution** When reading or writing to MIN while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 counter and 8.4.4 Writing to real-time clock 2 counter.

### 8.3.8 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 is a decimal counter that 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 2 clocks 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 does not clear this register to its default value.

**Figure 8-9. Format of Hour Count Register (HOUR)**

Address: FFF94H After reset: Undefined 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 reading or writing to HOUR while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 counter and 8.4.4 Writing to real-time clock 2 counter.

Table 8-3 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

**Table 8-3. Displayed Time Digits**

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 0)	
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.

### 8.3.9 Date 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 is a decimal counter that count ups when the hour counter overflows.

This counter counts as follows.

[DAY count values]

- 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 2 clocks 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 does not clear this register to its default value.

**Figure 8-10. Format of Day-of-week Count Register (DAY)**

Address: FFF96H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

**Caution** When reading or writing to DAY while the clock counter operates ( $RTCE = 1$ ), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 counter and 8.4.4 Writing to real-time clock 2 counter.

**8.3.10 Day-of-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 is a decimal counter that counts up when a carry to the date counter occurs.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks of `frtc` 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 does not clear this register to its default value.

**Figure 8-11. Format of Date Count Register (WEEK)**

Address: FFF95H After reset: Undefined 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 reading or writing to WEEK while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 counter and 8.4.4 Writing to real-time clock 2 counter.**

### 8.3.11 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 is a decimal counter that count ups when the date counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks of `fRTC` 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 does not clear this register to its default value.

**Figure 8-12. Format of Month Count Register (MONTH)**

Address: FFF97H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

**Caution** When reading or writing to MONTH while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 counter and 8.4.4 Writing to real-time clock 2 counter.

### 8.3.12 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 is a decimal counter that 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 2 clocks of `fRTC` 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 does not clear this register to its default value.

**Figure 8-13. Format of Year Count Register (YEAR)**

Address: FFF98H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

**Caution** When reading or writing to YEAR while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 counter and 8.4.4 Writing to real-time clock 2 counter.

**8.3.13 Clock error correction register (SUBCUD)**

This register is used to correct the clock with a minimum resolution and accuracy of 0.96 ppm when it is slow or fast by changing the counter value every second.

F8 to F0 of SUBCUD are 9-bit fixed-point (two's complement) register. For details, see **Table 8-5 Clock Error Correction Values**.

The SUBCUD register can be set by a 16-bit memory manipulation instruction.

Internal reset generated by the power-on-reset circuit clears this register to 0020H.

**Figure 8-14. Format of Clock Error Correction Register (SUBCUD)**

Address: F0310H After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUBCUD	F15	0	0	0	0	0	0	F8	F7	F6	F5	F4	F3	F2	F1	F0

F15	Clock error correction enable
0	Stops clock error correction.
1	Enables clock error correction.

The range of value that can be corrected by using the clock error correction register (SUBCUD) is shown in Table 8-4.

**Table 8-4. Correctable Range of Crystal Resonator Oscillation Frequency Deviation**

Item	Value
Correctable range	-274.6 ppm to +212.6 ppm
Maximum quantization error	±0.48 ppm
Minimum resolution	0.96 ppm



**Table 8-5. Clock Error Correction Values**

SUBCUD										Target Correction Values	
F15	F8	F7	F6	F5	F4	F3	F2	F1	F0		
1	1	0	0	0	0	0	0	0	0	-274.6 ppm	
	1	0	0	0	0	0	0	0	1	-273.7 ppm	
	1	0	0	0	0	0	0	1	0	-272.7 ppm	
	•	•	•	•	•	•	•	•	•	•	□
	•	•	•	•	•	•	•	•	•	•	□
	•	•	•	•	•	•	•	•	•	•	□
	1	1	1	1	1	1	1	1	0	1	-33.3 ppm
	1	1	1	1	1	1	1	1	1	0	-32.4 ppm
	1	1	1	1	1	1	1	1	1	1	-31.4 ppm
	0	0	0	0	0	0	0	0	0	0	-30.5 ppm
	0	0	0	0	0	0	0	0	0	1	-29.6 ppm
	0	0	0	0	0	0	0	0	1	0	-28.6 ppm
	•	•	•	•	•	•	•	•	•	•	□
	•	•	•	•	•	•	•	•	•	•	□
	•	•	•	•	•	•	•	•	•	•	□
	0	0	0	0	1	1	1	1	1	1	-0.95 ppm
	0	0	0	1	0	0	0	0	0	0	0 ppm
	0	0	0	1	0	0	0	0	0	1	0.95 ppm
	•	•	•	•	•	•	•	•	•	•	□
	•	•	•	•	•	•	•	•	•	•	□
•	•	•	•	•	•	•	•	•	•	□	
0	1	1	1	1	1	1	1	0	1	210.7 ppm	
0	1	1	1	1	1	1	1	1	0	211.7 ppm	
0	1	1	1	1	1	1	1	1	1	212.6 ppm	
0	x	x	x	x	x	x	x	x	x	Clock error correction stopped	

The F8 to F0 value of the SUBCUD register is calculated from the target correction value by using the following expression.

$$SUBCUD[8:0] = \left[ \frac{\text{Target correction value [ppm]} \times 2^{15}}{10^6} \right] \text{ 2's complement (9 bit fixed-point format)} + 0001.00000B$$

**Caution** The target correction value is the oscillation frequency deviation (unit: [ppm]) of the crystal resonator. For calculating the correction value, see 8.4.8 Example of watch error correction of real-time clock 2.

**Examples 1.** When target correction value = 18.3 [ppm]

$$\begin{aligned} SUBCUD[8:0] &= (18.3 \times 2^{15} / 10^6) \text{ 2's complement (9 bit fixed-point format)} + 0001.00000B \\ &= (0.59375) \text{ 2's complement (9 bit fixed-point format)} + 0001.00000B \\ &= 0000.10011B + 0001.00000B \\ &= 0001.10011B \end{aligned}$$

**Examples 2.** When target correction value = -18.3 [ppm]

$$\begin{aligned}\text{SUBCUD}[8:0] &= (-18.3 \times 2^{15} / 10^6) \text{ 2's complement (9 bit fixed-point format) } + 0001.00000\text{B} \\ &= (-0.59965) \text{ 2's complement (9 bit fixed-point format) } + 0001.00000\text{B} \\ &= 1111.01101\text{B} + 0001.00000\text{B} \\ &= 0000.01101\text{B}\end{aligned}$$

### 8.3.14 Alarm minute register (ALARMWM)

This register is used to set the minute of an alarm.

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

Reset signal generation does not clear this register to its default value.

**Figure 8-15. Format of Alarm Minute Register (ALARMWM)**

Address: FFF9AH After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

**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.

### 8.3.15 Alarm hour register (ALARMWH)

This register is used to set the hour of an alarm.

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

Reset signal generation does not clear this register to its default value.

**Figure 8-16. Format of Alarm Hour Register (ALARMWH)**

Address: FFF9BH After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

- Cautions**
1. Set a decimal value of 00 to 23 or 01 to 12 and 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.
  2. Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

**8.3.16 Alarm day-of-week register (ALARMWW)**

This register is used to set the day of the week of an alarm.

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

Reset signal generation does not clear this register to its default value.

**Figure 8-17. Format of Alarm Day-of-Week Register (ALARMWW)**

Address: FFF9CH After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Table 8-6 shows an example of setting the alarm.

**Table 8-6. Setting Alarm**

Time of Alarm	Day of the Week							12-Hour Display				24-Hour Display			
	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Hour	Hour	Min.	Min.	Hour	Hour	Min.	Min.
	W	W	W	W	W	W	W	10	1	10	1	10	1	10	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

### 8.3.17 Registers controlling port functions of real-time clock 2 output pins

When using real-time clock 2, set the registers that control the port functions multiplexed on the target channel (LCD port function registers 0 to 6 (PFSEG0 to PFSEG6), port mode register (PMxx), and port register (Pxx)). For details, see **4.3.9 LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)**, **4.3.1 Port mode registers (PMxx)**, and **4.3.2 Port registers (Pxx)**.

When using the ports (such as P31/RTC1HZ) to be shared with the real-time clock 2 output pins for real-time clock 2, set the LCD port function register (PFSEGx) bit, port mode register (PMxx) bit, and port register (Pxx) bit corresponding to each port to 0.

Example: When using P31/RTC1HZ for real-time clock 2 output

Set the PFSEG21 bit of LCD port function register (PFSEG2) to 0.

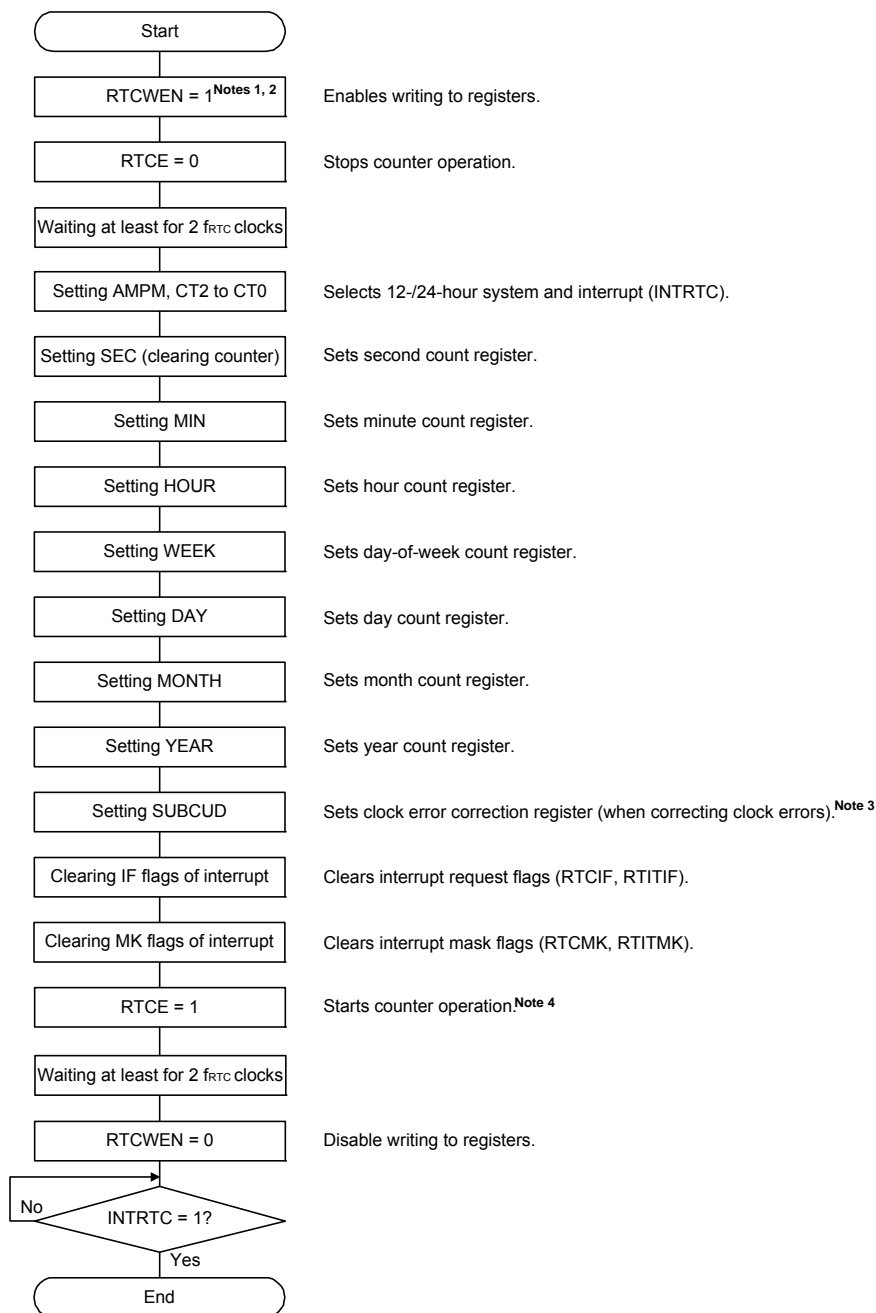
Set the PM31 bit of port mode register (PM3) to 0.

Set the P31 bit of port register (P3) to 0.

### 8.4 Real-time Clock 2 Operation

#### 8.4.1 Starting operation of real-time clock 2

Figure 8-18. Procedure for Starting Operation of Real-time Clock 2



- Notes**
1. Set RTCWEN to 0, except when accessing the RTC register, in order to prevent error when writing to the clock counter.
  2. First set the RTCWEN bit to 1, while oscillation of the count clock ( $f_{RTC}$ ) is stable.
  3. Set up the SUBCUD register only if the watch error must be corrected. For details about how to calculate the correction value, see **8.4.8 Example of watch error correction of real-time clock 2**.
  4. Confirm the procedure described in **8.4.2 Shifting to HALT/STOP mode after starting operation** when shifting to HALT/STOP mode without waiting for INTRTC = 1 after RTCE = 1.

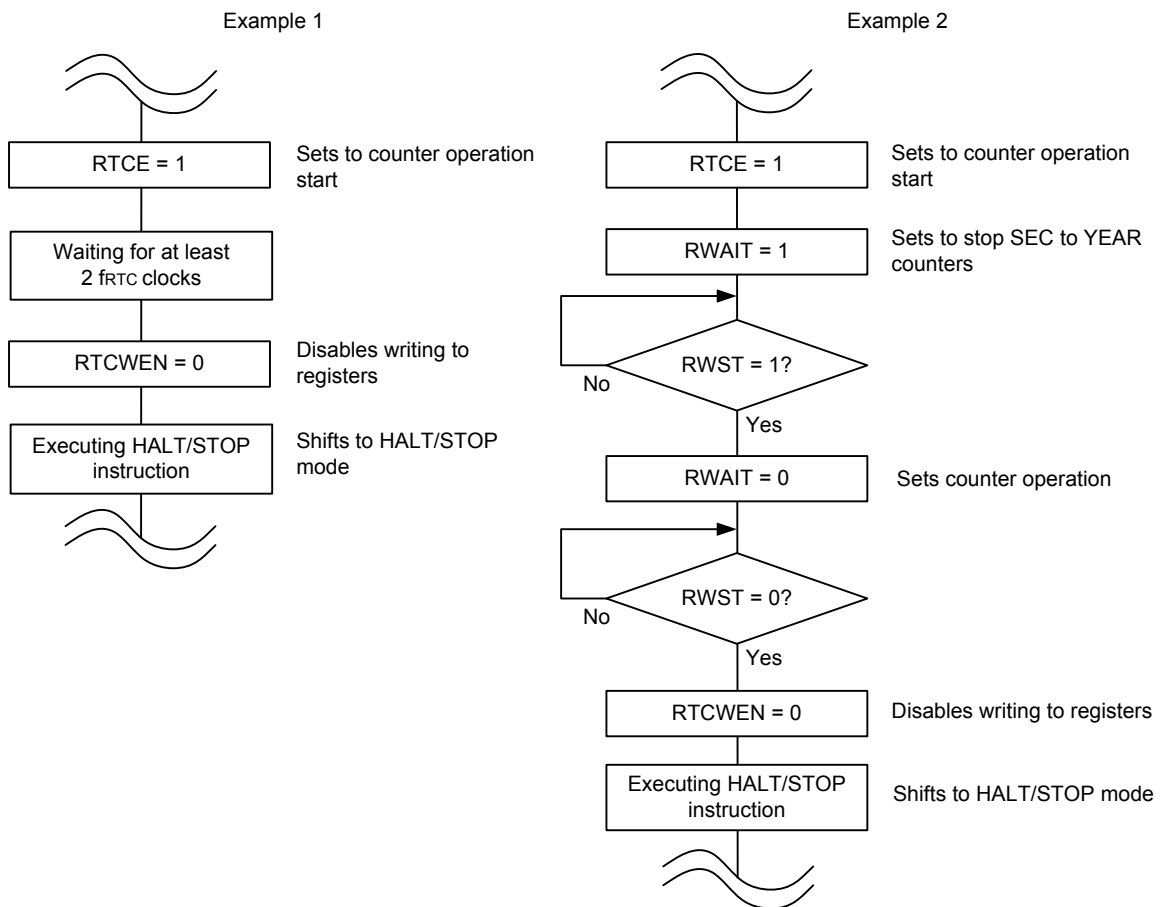
**8.4.2 Shifting to HALT/STOP mode after starting operation**

Perform one of the following processing when shifting to 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 STOP mode after the first INTRTC interrupt has occurred.

- (1) Shifting to HALT/STOP mode when at least two input clocks of the count clock (f<sub>RTC</sub>) have elapsed after setting the RTCE bit to 1 (see Example 1 of **Figure 8-19**).
- (2) 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 Example 2 of **Figure 8-19**).

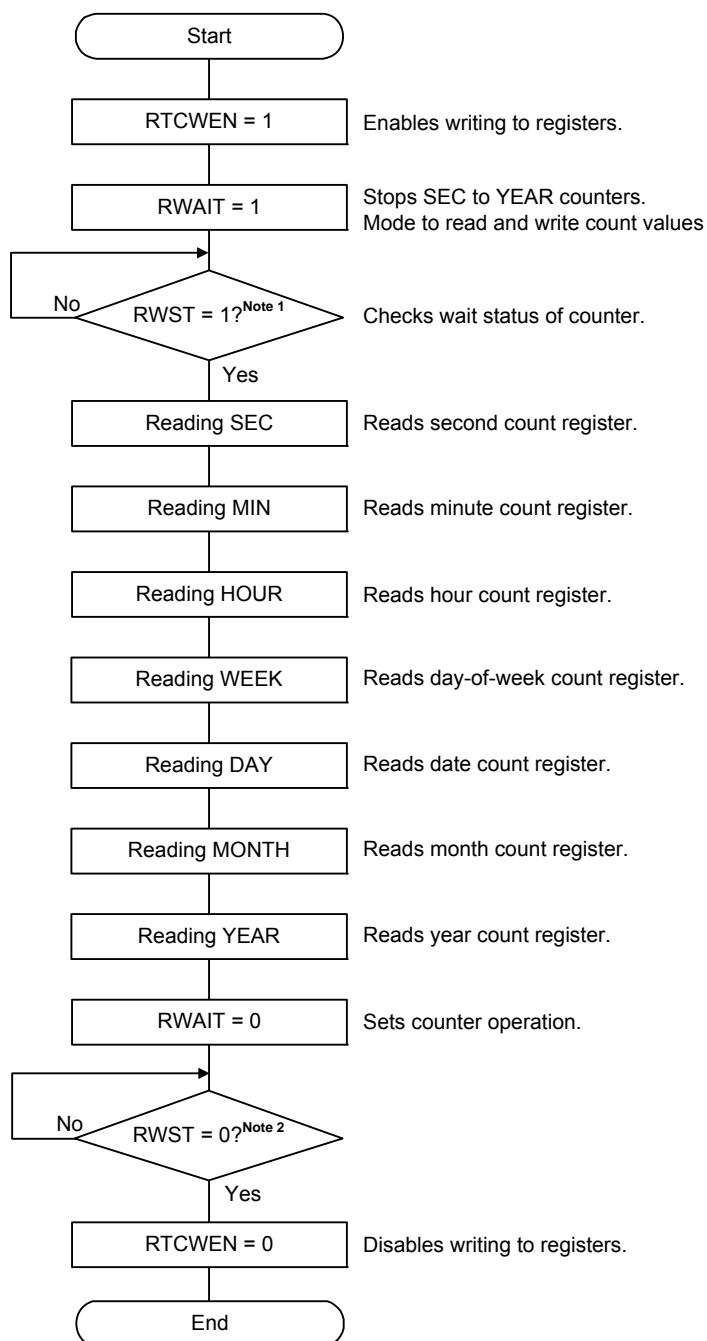
**Figure 8-19. Procedure for Shifting to HALT/STOP Mode after Setting RTCE = 1**



### 8.4.3 Reading real-time clock 2 counter

Read the counter after setting RWAIT to 1.  
Set RWAIT to 0 after completion of reading the counter.

**Figure 8-20. Procedure for Reading Real-time Clock 2**



- Notes**
1. When the counter is stopped (RTCE = 0), RWST is not set to 1.
  2. Be sure to confirm that RWST = 0 before shifting to STOP mode.

**Caution** Complete setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

**Remark** SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence. All the registers do not have to be set and only some registers may be read.

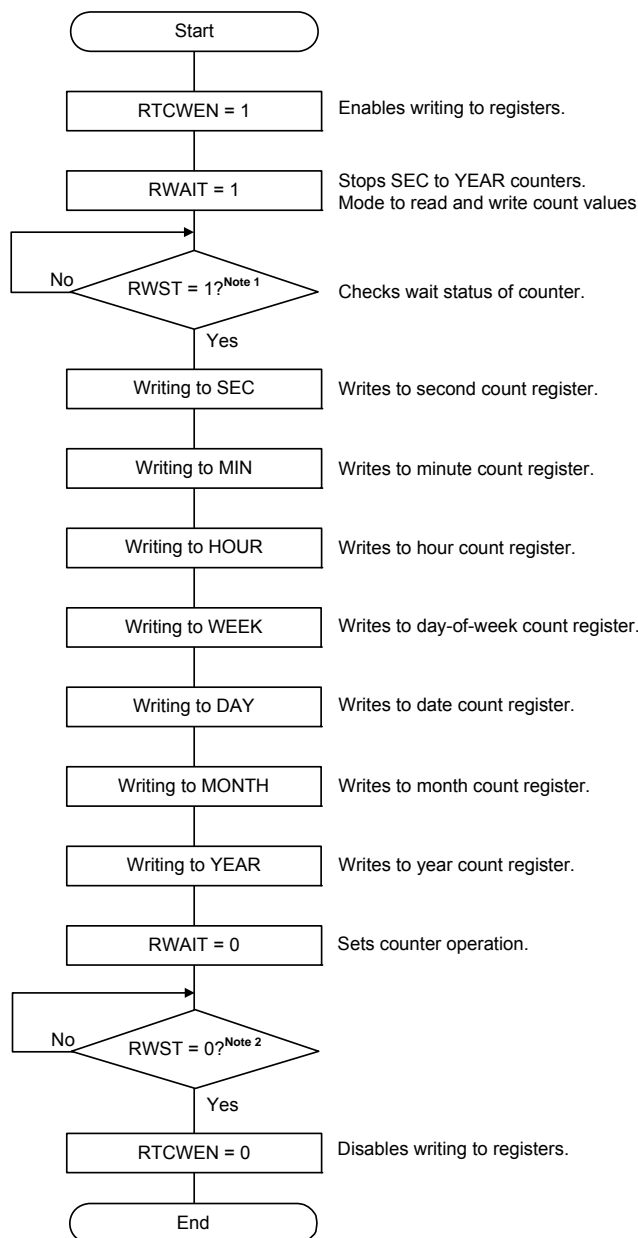


### 8.4.4 Writing to real-time clock 2 counter

Write to the counter after setting RWAIT to 1.

Set RWAIT to 0 after completion of writing the counter.

**Figure 8-21. Procedure for Writing Real-time Clock 2 Counter**



- Notes**
1. When the counter is stopped (RTCE = 0), RWST is not set to 1.
  2. Be sure to confirm that RWST = 0 before shifting to STOP mode.

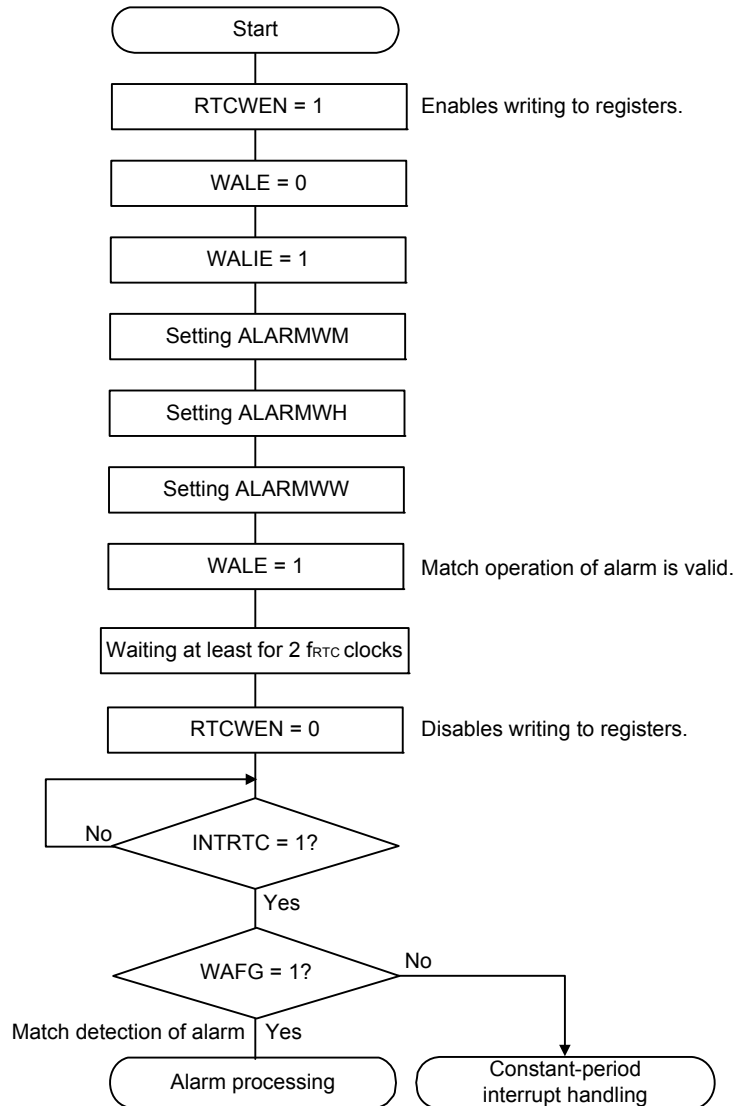
- Cautions**
1. Complete setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.
  2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR registers while the counter operates (RTCE = 1), rewrite the values of the MIN register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG, and RTCIF flags after rewriting the MIN register.

**Remark** SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence. All the registers do not have to be set and only some registers may be written.

8.4.5 Setting alarm of real-time clock 2

Set the alarm time after setting WALE to 0 (to disable alarm operation).

Figure 8-22. Alarm Setting Procedure

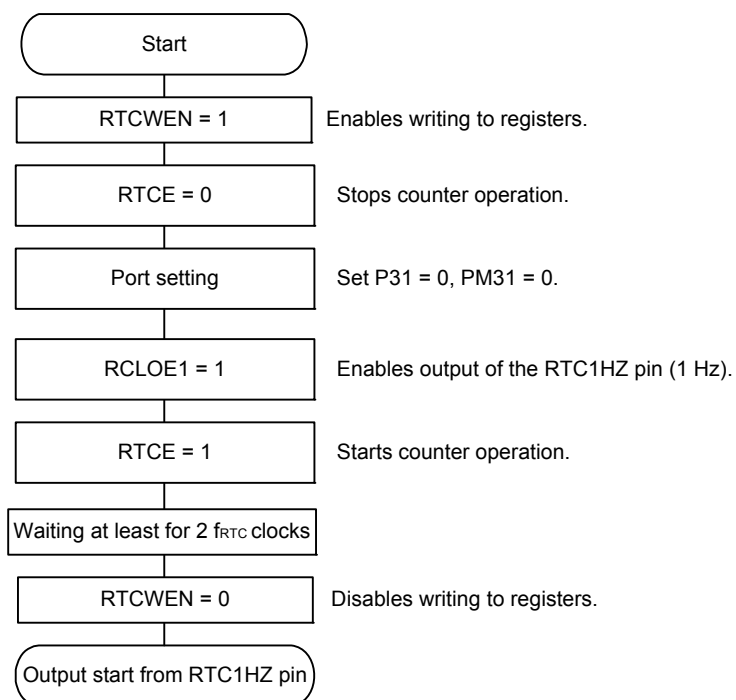


**Remarks 1.** ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

**2.** Constant-period 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 constant-period interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

## 8.4.6 1 Hz output of real-time clock 2

Figure 8-23. 1 Hz Output Setting Procedure

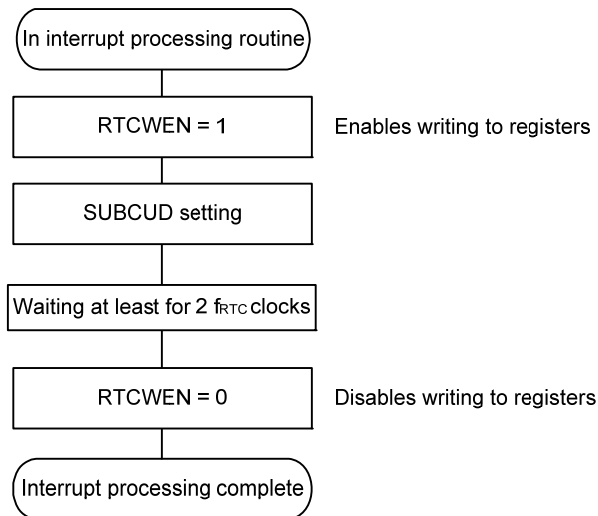


**8.4.7 Clock error correction register setting procedure**

RTC correction may not be successful if there is a conflict between the clock error correction register (SUBCUD) rewrite and correction timing. In order to prevent conflict between the correction timing and rewrite of the SUBCUD register, be sure to complete rewrite of the SUBCUD register before the next correction timing occurs (within approx. 0.5 seconds), which is calculated starting from the correction timing interrupt (INTRTIT) or periodic interrupt (INTRTC) that is synchronized with the correction timing.

- Set the clock error correction register after setting RTCWEN to 1. Set RTCWEN to 0 after completion of register setting.

**Figure 8-24. Clock Error Correction Register Setting Procedure**



### 8.4.8 Example of watch error correction of real-time clock 2

The clock can be corrected every second with a minimum resolution and accuracy of 0.96 ppm when it is slow or fast, by setting a value to the clock error correction register.

The following shows how to calculate the target correction value, and how to calculate the F8 to F0 values of the clock error correction register from the target correction value.

#### Calculating the target correction value

(Output frequency of the RTC1HZ pin using)

#### [Measuring the oscillation frequency]

The oscillation frequency<sup>Note</sup> of each product is measured by outputting 1 Hz from the RTC1HZ pin when the F15 bit of the watch error correction register (SUBCUD) is cleared to 0 (stops the watch error correction).

**Note** See 8.4.6 1 Hz output of real-time clock 2 for the procedure of outputting about 1 Hz from the RTC1HZ pin.

#### [Calculating the target correction value]

(When the output frequency from the RTCCL pin is 0.9999817 Hz)

$$\text{Oscillation frequency} = 32768 \times 0.9999817 \approx 32767.40 \text{ Hz}$$

Assume the target frequency to be 32768 Hz. Then the target correction value is calculated as follows.

$$\begin{aligned} \text{Target correction value} &= (\text{Oscillation frequency} - \text{Target frequency}) \div \text{Target frequency} \\ &= (32767.40 - 32768.00) \div 32768.00 \\ &\approx -18.3 \text{ ppm} \end{aligned}$$

- Remarks**
1. The oscillation frequency is the frequency of the input clock ( $f_{\text{RTC}}$ ). It can be calculated from the output frequency of the RTC1HZ pin  $\times$  32768 when watch error correction is not operating.
  2. The target correction value is the oscillation frequency deviation (unit: [ppm]) of the crystal resonator.
  3. The target frequency is the frequency resulting after watch error correction performed.

**Calculating the F8 to F0 value of the watch error correction register**

The F8 to F0 values in the SUBCUD register is calculated from the target correction value by using the following expression.

$$\text{SUBCUD}[8:0] = \left[ \frac{\text{Target correction value [ppm]} \times 2^{15}}{10^6} \right]_{2\text{'s complement (9 bit fixed-point format)}} + 0001.00000\text{B}$$

**Examples 1.** When target correction value = -18.3 [ppm]

$$\begin{aligned} \text{SUBCUD}[8:0] &= (-18.3 \times 2^{15} / 10^6)_{2\text{'s complement (9 bit fixed-point format)}} + 0001.00000\text{B} \\ &= (-0.59965)_{2\text{'s complement (9 bit fixed-point format)}} + 0001.00000\text{B} \\ &= 1111.01101\text{B} + 0001.00000\text{B} \\ &= 0000.01101\text{B} \end{aligned}$$

**Examples 2.** When target correction value = 94.0 [ppm]

$$\begin{aligned} \text{SUBCUD}[8:0] &= (94.0 \times 2^{15} / 10^6)_{2\text{'s complement (9 bit fixed-point format)}} + 0001.00000\text{B} \\ &= (+3.08019)_{2\text{'s complement (9 bit fixed-point format)}} + 0001.00000\text{B} \\ &= 0011.00011\text{B} + 0001.00000\text{B} \\ &= 0100.00011\text{B} \end{aligned}$$

CHAPTER 9 12-BIT INTERVAL TIMER

9.1 Functions of 12-bit Interval Timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be used for waking the system up from STOP mode and triggering an A/D converter's SNOOZE mode.

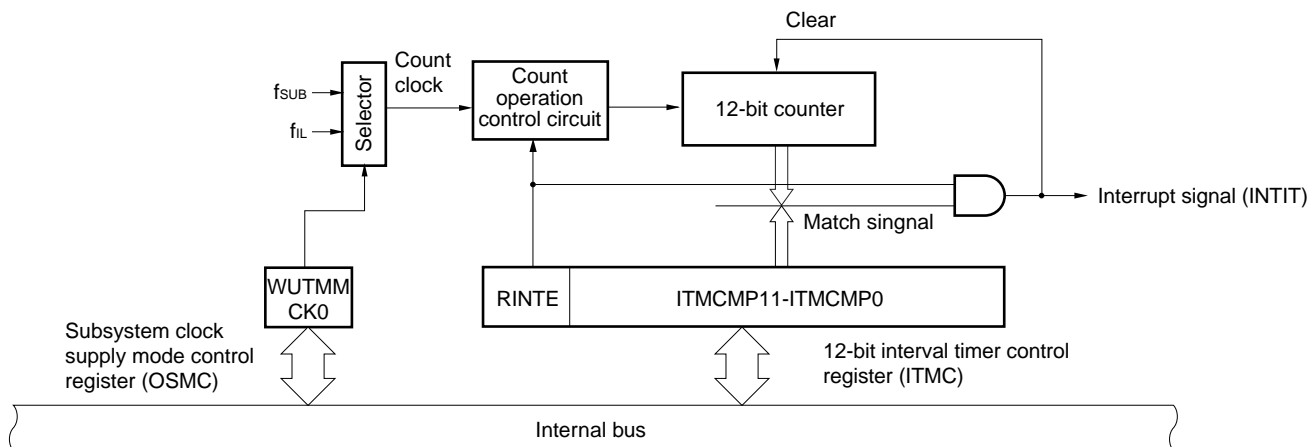
9.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 9-1. Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 1 (PER1)
	Subsystem clock supply mode control register (OSMC)
	12-bit interval timer control register (ITMC)

Figure 9-1. Block Diagram of 12-bit Interval Timer



### 9.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 1 (PER1)
- Subsystem clock supply mode control register (OSMC)
- 12-bit interval timer control register (ITMC)

#### 9.3.1 Peripheral enable register 1 (PER1)

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 (TMKAEN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-2. Format of Peripheral Enable Register 1 (PER1)**

Address: F007AH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
PER1	TMKAEN	0	CMPEN	TKB2EN	0	0	0	0

TMKAEN	Control of 12-bit interval timer input clock supply
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFRs used by the 12-bit interval timer cannot be written.</li> <li>• The 12-bit interval timer is in the reset status.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>• SFRs used by the 12-bit interval timer can be read and written.</li> </ul>

- Cautions**
1. When using the 12-bit interval timer, first set the TMKAEN bit to 1. If TMKAEN = 0, writing to a control register of 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 real-time clock 2, 12-bit interval timer, clock output/buzzer output, and LCD controller/driver can be stopped in 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 TMKAEN bit of the PER1 register to 1 and the other bits (bits 0 to 6) to 0.
  3. Be sure to clear the bits 0 to 3 and 6 to 0.



### 9.3.2 Subsystem clock supply mode control register (OSMC)

The OSMC 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 real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver, 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 operation clock of real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver.

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

Reset signal generation clears this register to 00H.

**Figure 9-3. 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

RTCLPC	Setting in STOP mode or in HALT mode while subsystem clock is selected as CPU clock
0	Enables supplying the subsystem clock to peripheral functions (See <b>Tables 21-1 to 21-3</b> for peripheral functions whose operations are enabled.)
1	Stops supplying the subsystem clock to peripheral functions other than real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver.

WUTMMCK0	Selection of operation clock for real-time clock 2, 12-bit interval timer, and LCD controller/driver	Selection of clock output from PCLBUZn pin of clock output/buzzer output controller
0	Subsystem clock ( $f_{SUB}$ )	Selecting the subsystem clock ( $f_{SUB}$ ) is enabled.
1	Low-speed on-chip oscillator clock ( $f_{IL}$ )	Selecting the subsystem clock ( $f_{SUB}$ ) is disabled.

- Cautions**
1. Be sure to select the subsystem clock (WUTMMCK0 bit = 0) if the subsystem clock is oscillating.
  2. When WUTMMCK0 is set to 1, the low-speed on-chip oscillator clock oscillates.
  3. The subsystem clock and low-speed on-chip oscillator clock can only be switched by using the WUTMMCK0 bit if real-time clock 2, 12-bit interval timer, and LCD controller/driver are all stopped.

**9.3.3 12-bit 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 sets this register to 0FFFH.

**Figure 9-4. Format of 12-bit 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	ITMCMP11 to ITMCMP0

RINTE	12-bit Interval timer operation control
0	Count operation stopped (count clear)
1	Count operation started

ITMCMP11 to ITMCMP0	Specification of the 12-bit interval timer compare value
001H	These bits generate an interrupt at the fixed cycle (count clock cycles x (ITMCMP setting + 1)).
•	
•	
FFFH	
000H	Setting prohibited

Example interrupt cycles when 001H or FFFH is specified for ITMCMP11 to ITMCMP0	
•	ITMCMP11 to ITMCMP0 = 001H, count clock: when $f_{SUB} = 32.768 \text{ kHz}$ $1/32.768 \text{ [kHz]} \times (1 + 1) = 0.06103515625 \text{ [ms]} \cong 61.03 \text{ [}\mu\text{s]}$
•	ITMCMP11 to ITMCMP0 = FFFH, count clock: when $f_{SUB} = 32.768 \text{ kHz}$ $1/32.768 \text{ [kHz]} \times (4095 + 1) = 125 \text{ [ms]}$

- 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 (changing the RINTE bit from 0 to 1) again, clear the TMKAIF 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 ITMCMP11 to ITMCMP0 bits when RINTE = 0. However, it is possible to change the settings of the ITMCMP11 to ITMCMP0 bits at the same time as when changing RINTE from 0 to 1 or 1 to 0.

### 9.4 12-bit Interval Timer Operation

#### 9.4.1 12-bit interval timer operation timing

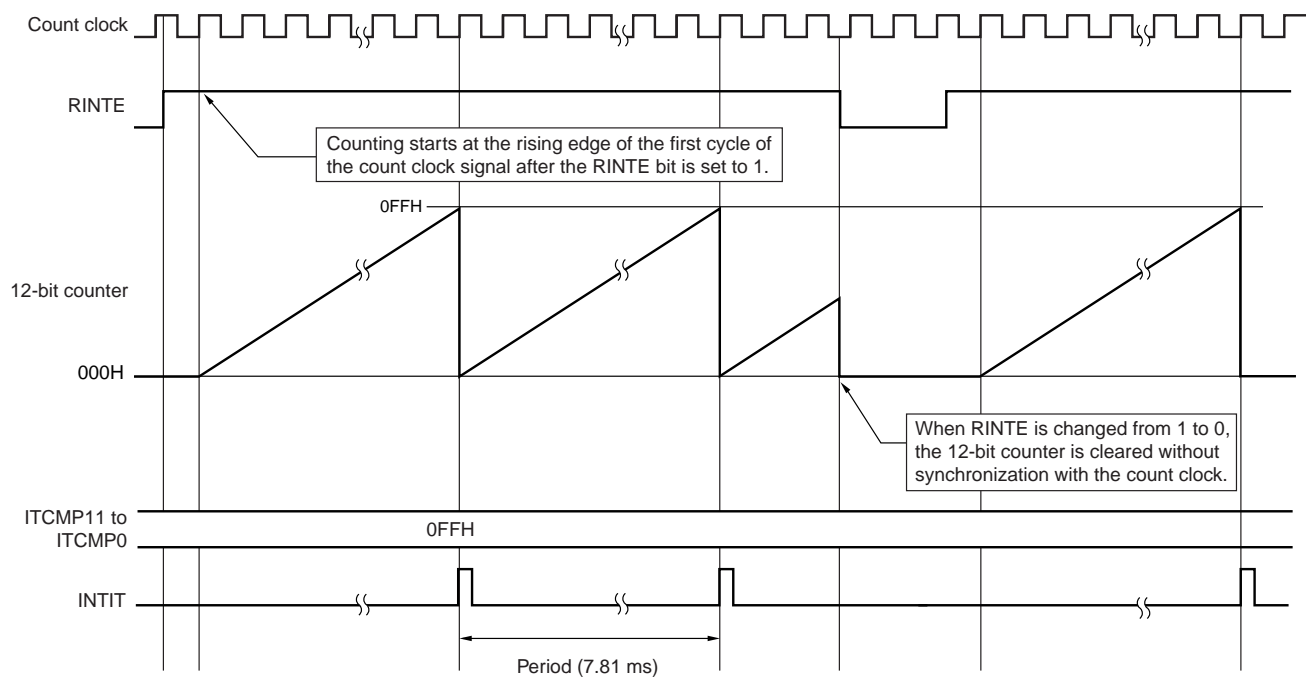
The count value specified for the ITMCMP11 to ITMCMP0 bits is used as an interval to operate a 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

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 ITMCMP11 to ITMCMP0 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 9-5. 12-bit Interval Timer Operation Timing (ITMCMP11 to ITMCMP0 = 0FFH, count clock:  $f_{SUB} = 32.768$  kHz)**

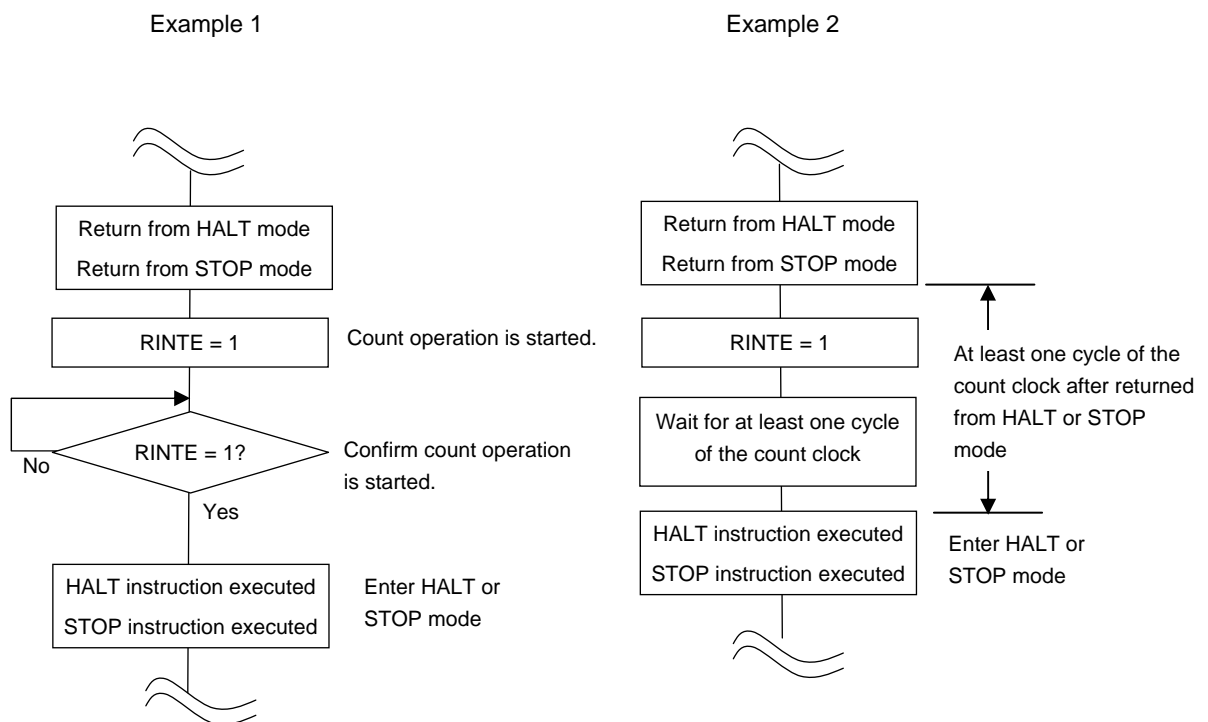


**9.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode**

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 9-6).
- 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 9-6).

**Figure 9-6. Procedure of Entering to HALT or STOP Mode After Setting RINTE to 1**



## CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

### 10.1 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.

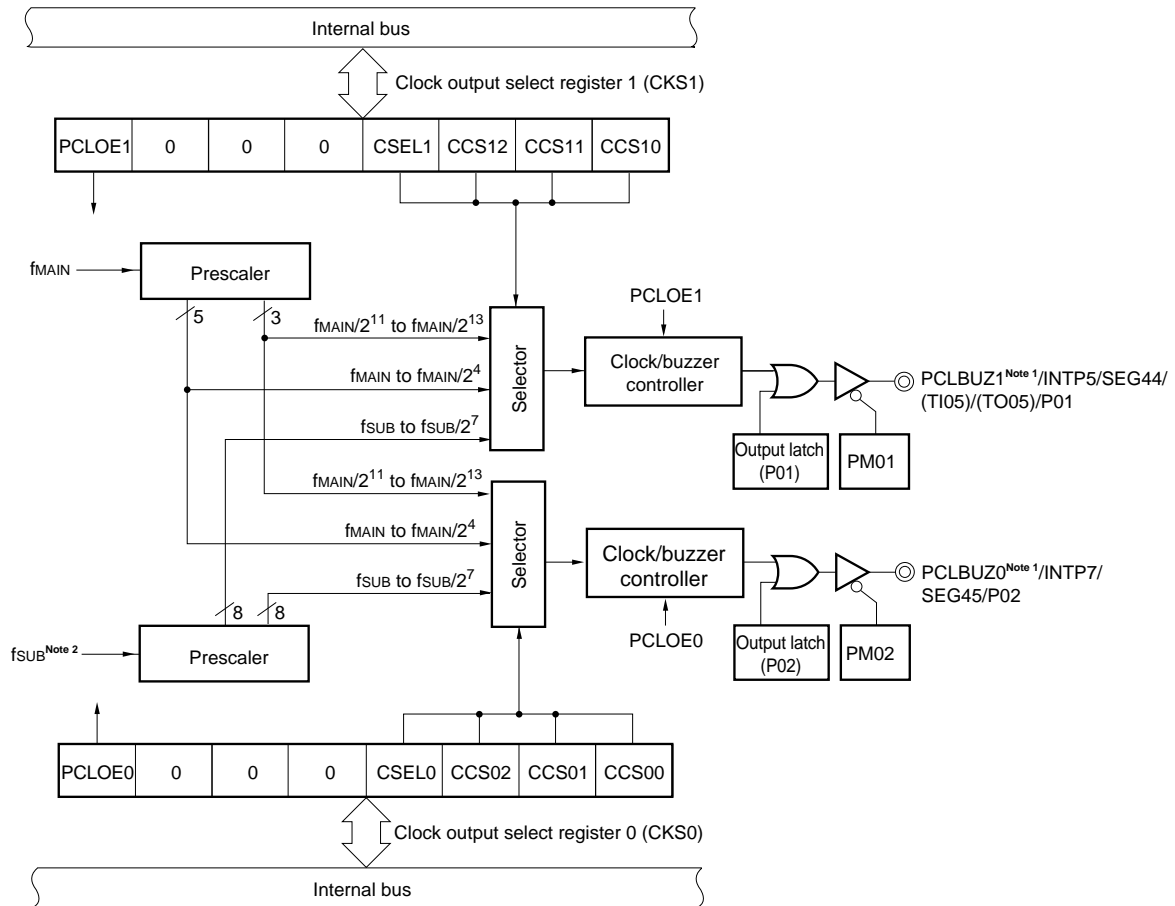
Two output pins, PCLBUZ0 and PCLBUZ1, are available.

The PCLBUZn pin outputs a clock selected by clock output select register n (CKSn).

Figure 10-1 shows the block diagram of clock output/buzzer output controller.

**Remark** n = 0, 1

Figure 10-1. Block Diagram of Clock Output/Buzzer Output Controller



- Notes**
1. For the frequencies that can be output from PCLBUZ0 and PCLBUZ1, see **32.4 AC Characteristics** or **33.4 AC Characteristics**.
  2. Selecting  $f_{SUB}$  as the output clock of the clock output/buzzer output controller is prohibited when the WUTMMCK0 bit of the OSMC register is set to 1.

**Remark** The above diagram shows the clock output/buzzer output pins when PIOR3 is 0. In other cases, the name of pins, output latches (Pxx) and PMxx should be read differently.

## 10.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

**Table 10-1. Configuration of Clock Output/Buzzer Output Controller**

Item	Configuration
Control registers	Clock output select register n (CKSn) Subsystem clock supply mode control register (OSMC) Port mode register 0 (PM0) Port register 0 (P0)

## 10.3 Registers Controlling Clock Output/Buzzer Output Controller

The following two registers are used to control the clock output/buzzer output controller.

- Clock output select register n (CKSn)
- Subsystem clock supply mode control register (OSMC)
- Port mode register 0 (PM0)

### 10.3.1 Clock output select register n (CKSn)

This register specifies output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZn), and specifies the output clock.

Select the clock to be output from the PCLBUZn pin by using the CKSn register.

The CKSn register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-2. 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 <sub>MAIN</sub> = 5 MHz	f <sub>MAIN</sub> = 10 MHz	f <sub>MAIN</sub> = 20 MHz	f <sub>MAIN</sub> = 24 MHz
0	0	0	0	f <sub>MAIN</sub>	5 MHz	10 MHz <sup>Note 1</sup>	Setting prohibited <sup>Note 1</sup>	Setting prohibited <sup>Note 1</sup>
0	0	0	1	f <sub>MAIN</sub> /2	2.5 MHz	5 MHz	10 MHz <sup>Note 1</sup>	12 MHz <sup>Note 1</sup>
0	0	1	0	f <sub>MAIN</sub> /2 <sup>2</sup>	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	f <sub>MAIN</sub> /2 <sup>3</sup>	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	f <sub>MAIN</sub> /2 <sup>4</sup>	312.5 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	f <sub>MAIN</sub> /2 <sup>11</sup>	2.44 kHz	4.88 kHz	9.76 kHz	11.7 kHz
0	1	1	0	f <sub>MAIN</sub> /2 <sup>12</sup>	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
0	1	1	1	f <sub>MAIN</sub> /2 <sup>13</sup>	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	0	0	0	f <sub>SUB</sub> <sup>Note 2</sup>	32.768 kHz			
1	0	0	1	f <sub>SUB</sub> /2 <sup>Note 2</sup>	16.384 kHz			
1	0	1	0	f <sub>SUB</sub> /2 <sup>2</sup> <sup>Note 2</sup>	8.192 kHz			
1	0	1	1	f <sub>SUB</sub> /2 <sup>3</sup> <sup>Note 2</sup>	4.096 kHz			
1	1	0	0	f <sub>SUB</sub> /2 <sup>4</sup> <sup>Note 2</sup>	2.048 kHz			
1	1	0	1	f <sub>SUB</sub> /2 <sup>5</sup> <sup>Note 2</sup>	1.024 kHz			
1	1	1	0	f <sub>SUB</sub> /2 <sup>6</sup> <sup>Note 2</sup>	512 Hz			
1	1	1	1	f <sub>SUB</sub> /2 <sup>7</sup> <sup>Note 2</sup>	256 Hz			

- Notes**
1. Use the output clock up to 16 MHz. See 32.4 AC Characteristics or 33.4 AC Characteristics for details.
  2. Selecting f<sub>SUB</sub> as the output clock of the clock output/buzzer output controller is prohibited when the WUTMMCK0 bit of the OSMC register is set to 1.

- 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 to 0 before executing the STOP instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn can be set to 1 because the clock can be output in STOP mode.

- Remarks**
1. n = 0, 1
  2. f<sub>MAIN</sub>: Main system clock frequency  
f<sub>SUB</sub>: Subsystem clock frequency



### 10.3.2 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 other than real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver is stopped in STOP mode or in HALT mode while the subsystem clock is selected as the CPU clock.

In addition, the OSMC register can be used to select the operation clock of real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver.

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

Reset signal generation clears this register to 00H.

**Figure 10-3. 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

RTCLPC	Setting in STOP mode or in HALT mode while subsystem clock is selected as CPU clock
0	Enables supplying the subsystem clock to peripheral functions (See <b>Tables 21-1 to 21-3</b> for peripheral functions whose operations are enabled.)
1	Stops supplying the subsystem clock to peripheral functions other than real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver.

WUTMMCK0	Selection of operation clock for real-time clock 2, 12-bit interval timer, and LCD controller/driver	Selection of clock output from PCLBUZn pin of clock output/buzzer output controller
0	Subsystem clock ( $f_{SUB}$ )	Selecting the subsystem clock ( $f_{SUB}$ ) is enabled.
1	Low-speed on-chip oscillator clock ( $f_{IL}$ )	Selecting the subsystem clock ( $f_{SUB}$ ) is disabled.

- Cautions**
1. Be sure to select the subsystem clock (WUTMMCK0 bit = 0) if the subsystem clock is oscillating.
  2. When WUTMMCK0 is set to 1, the low-speed on-chip oscillator clock oscillates.
  3. Do not select  $f_{SUB}$  as the clock output or buzzer output clock when the WUTMMCK0 bit is 1.

### 10.3.3 Registers controlling port functions of clock output/buzzer output pins

When using the clock output/buzzer output function, set the registers that control the port functions multiplexed on the target channel (port mode register (PMxx), port register (Pxx)). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

When using a port pin with a multiplexed clock output/buzzer output pins (e.g. P02/SEG45/INTP7/PCLBUZ0, P01/SEG44(TI05)/(TO05)/(INTP5)/PCLBUZ1) for clock output/buzzer output, set the corresponding bits in the port mode register (PMxx) and port register (Pxx) to 0.

Example: When P02/SEG45/INTP7/PCLBUZ0 is to be used for clock output/buzzer output  
Set the PM02 bit of port mode register 0 to 0.  
Set the P02 bit of port register 0 to 0.

## 10.4 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 clock output select register 0 (CKS0).

The PCLBUZ1 pin outputs a clock/buzzer selected by clock output select register 1 (CKS1).

### 10.4.1 Operation as output pin

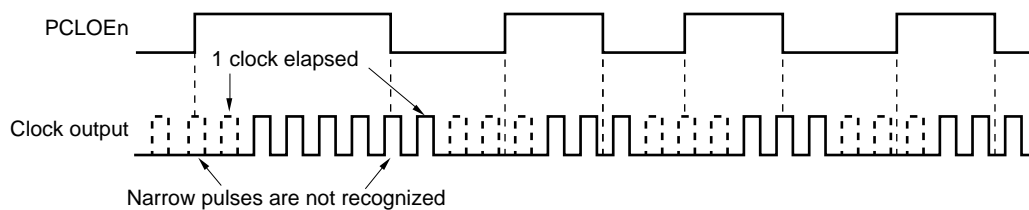
Use the following procedure to output a clock or buzzer from the PCLBUZn pin.

- <1> Set 0 in the bit of the port mode register (PMxx) and port register (Pxx) which correspond to the port which has a pin used as the PCLBUZ0 pin.
- <2> Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of clock output select register n (CKSn) for the PCLBUZn pin (output is disabled).
- <3> Set bit 7 (PCLOEn) of the CKSn register to 1 to enable clock/buzzer output.

**Remarks 1.** The clock output controller starts or stops outputting the clock one cycle after enabling or disabling clock output (PCLOEn bit). At this time, pulses with a narrow width are not output. Figure 10-4 shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.

2. n = 0, 1

**Figure 10-4. Timing of Outputting Clock from PCLBUZn Pin**



## 10.5 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 main system clock cycles after the output is disabled (PCLOEn = 0), the PCLBUZn output width becomes shorter.

## CHAPTER 11 WATCHDOG TIMER

### 11.1 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.

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 22 RESET FUNCTION**.

When  $75\% + 1/2/f_{IL}$  of the overflow time is reached, an interval interrupt can be generated.

### 11.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

**Table 11-1. Configuration of Watchdog Timer**

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

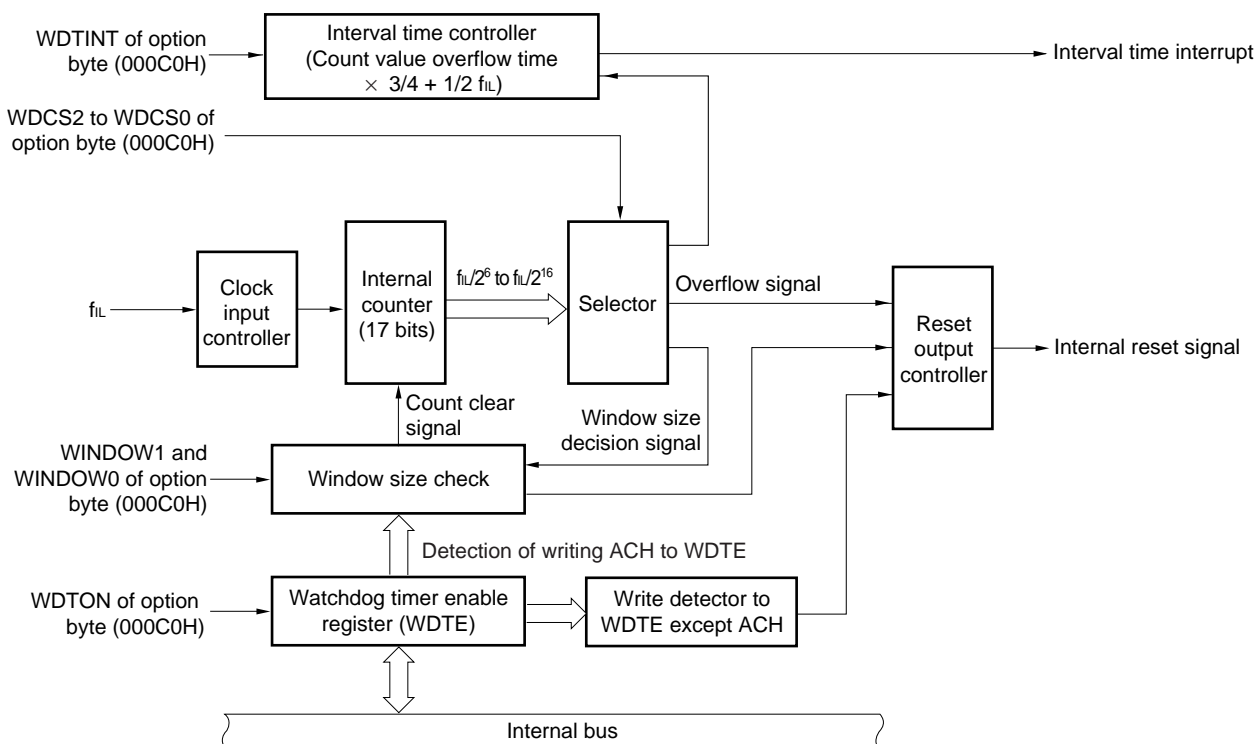
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

**Table 11-2. 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 27 OPTION BYTE**.

**Figure 11-1. Block Diagram of Watchdog Timer**



**Remark**  $f_{IL}$ : Low-speed on-chip oscillator clock

### 11.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

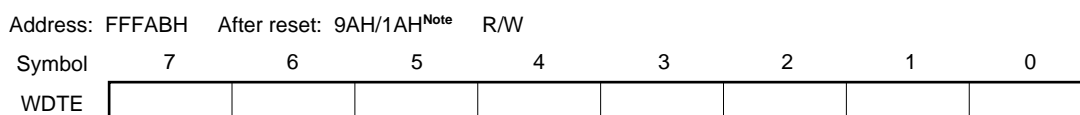
#### 11.3.1 Watchdog timer enable register (WDTE)

Writing “ACH” to the WDTE register clears the watchdog timer counter and the watchdog timer 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<sup>Note</sup>.

**Figure 11-2. Format of Watchdog Timer Enable Register (WDTE)**



**Note** The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

- Cautions**
1. If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.
  2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.
  3. The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).

## 11.4 Operation of Watchdog Timer

### 11.4.1 Controlling operation of watchdog timer

- 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 27**).

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 **11.4.2** and **CHAPTER 27**).
  - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **11.4.3** and **CHAPTER 27**).
- After a reset release, the watchdog timer starts counting.
  - 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.
  - 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.
  - 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**
- 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.
  - After "ACH" is written to the WDTE register, an error of up to 2 clocks (f<sub>IL</sub>) may occur before the watchdog timer is cleared.
  - 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.

**11.4.2 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 11-3. Setting of Overflow Time of Watchdog Timer**

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f <sub>IL</sub> = 17.25 kHz (MAX.))
0	0	0	2 <sup>6</sup> /f <sub>IL</sub> (3.71 ms)
0	0	1	2 <sup>7</sup> /f <sub>IL</sub> (7.42 ms)
0	1	0	2 <sup>8</sup> /f <sub>IL</sub> (14.84 ms)
0	1	1	2 <sup>9</sup> /f <sub>IL</sub> (29.68 ms)
1	0	0	2 <sup>11</sup> /f <sub>IL</sub> (118.72 ms)
1	0	1	2 <sup>13</sup> /f <sub>IL</sub> (474.89 ms)
1	1	0	2 <sup>14</sup> /f <sub>IL</sub> (949.79 ms)
1	1	1	2 <sup>16</sup> /f <sub>IL</sub> (3799.18 ms)

**Remark** f<sub>IL</sub>: Low-speed on-chip oscillator clock frequency

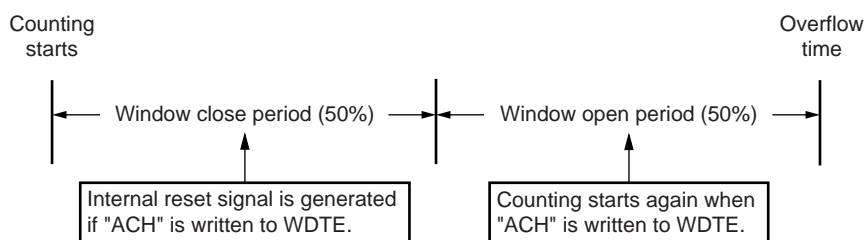


### 11.4.3 Setting window open period of watchdog timer

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 11-4. 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%

**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:  
 $2^9/f_{IL} \text{ (MAX.)} = 2^9/17.25 \text{ kHz} = 29.68 \text{ ms}$
- Window close time:  
 $0 \text{ to } 2^9/f_{IL} \text{ (MIN.)} \times (1 - 0.5) = 0 \text{ to } 2^9/12.75 \text{ kHz} \times 0.5 = 0 \text{ to } 20.08 \text{ ms}$
- Window open time:  
 $2^9/f_{IL} \text{ (MIN.)} \times (1 - 0.5) \text{ to } 2^9/f_{IL} \text{ (MAX.)} = 2^9/12.75 \text{ kHz} \times 0.5 \text{ to } 2^9/17.25 \text{ kHz} = 20.08 \text{ to } 29.68 \text{ ms}$

#### 11.4.4 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 11-5. 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 12 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

	64-pin	80-pin
Analog input channels	9 ch (ANI0, ANI1, ANI16, ANI20 to ANI25)	12 ch (ANI0, ANI1, ANI16 to ANI25)

## 12.1 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 12 channels of A/D converter analog inputs (ANI0, ANI1 and ANI16 to ANI25). 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.

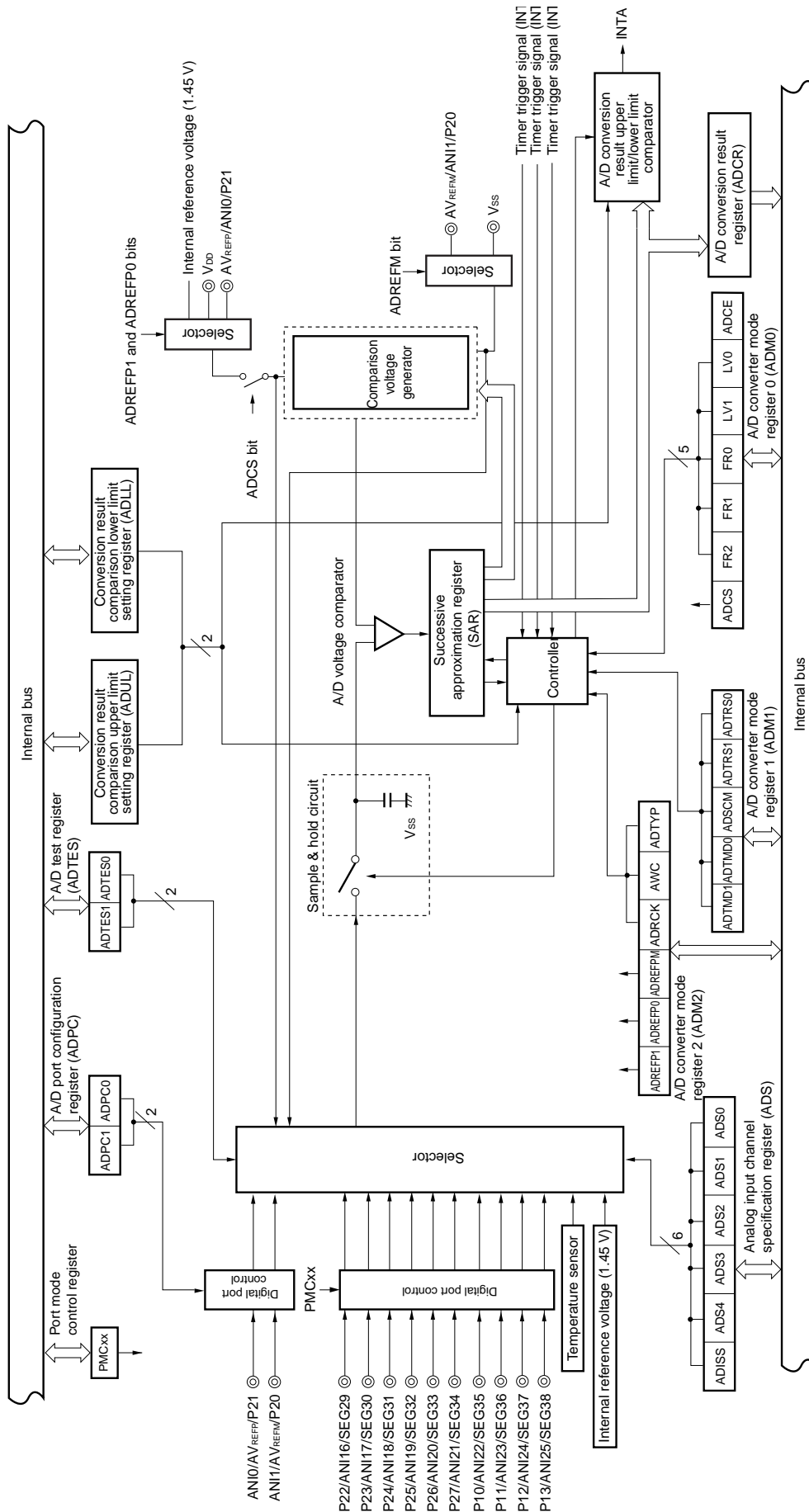
- **10-bit/8-bit resolution A/D conversion**

10-bit or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0, ANI1 and ANI16 to ANI25. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

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.
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 $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ .
	Low voltage 1 or low voltage 2 mode	Conversion is done in the operation voltage range of $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ . 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 12-1. Block Diagram of A/D Converter



**Remark** Analog input pin for figure 12-1 when a 80-pin product is used.

## 12.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

### (1) ANI0, ANI1 and ANI16 to ANI25 pins

These are the analog input pins of the 12 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  $\geq$  Voltage tap of comparison voltage generator: Bit 8 = 1

Analog input voltage  $\leq$  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<sub>REFP</sub> pin**

This pin inputs an external reference voltage (AV<sub>REFP</sub>).

If using AV<sub>REFP</sub> 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, ANI1 and ANI16 to ANI25 are converted to digital signals based on the voltage applied between AV<sub>REFP</sub> and the – side reference voltage (AV<sub>REFM</sub>/V<sub>SS</sub>).

In addition to AV<sub>REFP</sub>, it is possible to select V<sub>DD</sub> or the internal reference voltage (1.45 V) as the + side reference voltage of the A/D converter.

**(10) AV<sub>REFM</sub> pin**

This pin inputs an external reference voltage (AV<sub>REFM</sub>). If using AV<sub>REFM</sub> as the – side reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.

In addition to AV<sub>REFM</sub>, it is possible to select V<sub>SS</sub> as the – side reference voltage of the A/D converter.

### 12.3 Registers Used in A/D Converter

The A/D converter uses 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 1, 2 (PMC1, PMC2)
- Port mode registers 1, 2 (PM1, PM2)

### 12.3.1 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 12-2. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by the A/D converter cannot be written.</li> <li>• The A/D converter is in the reset status.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>• SFR used by the A/D converter can be read/written.</li> </ul>

- 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 1, 2 (PM1, PM2), port mode control registers 1, 2 (PMC1, PMC2), 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 bits 1 and 6 to 0.



**12.3.2 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 12-3. Format of A/D Converter Mode Register 0 (ADM0)**

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM0	ADCS	0	FR2 <sup>Note 1</sup>	FR1 <sup>Note 1</sup>	FR0 <sup>Note 1</sup>	LV1 <sup>Note 1</sup>	LV0 <sup>Note 1</sup>	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation [When read] Conversion stopped/standby status
1	Enables conversion operation [When read] 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

ADCE	A/D voltage comparator operation control <sup>Note 2</sup>
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

- Notes**
- For details of the FR2 to FR0, LV1, and LV0 bits, and A/D conversion, see **Table 12-3 A/D Conversion Time Selection**.
  - 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**
- Change the FR2 to FR0, LV1, and LV0 bits while conversion is stopped (ADCS = 0, ADCE = 0).
  - Do not set ADCS to 1 and ADCE to 0.
  - Do not change the ADCE and ADCS 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 12.7 A/D Converter Setup Flowchart.
  - Be sure to clear bit 6 to 0.

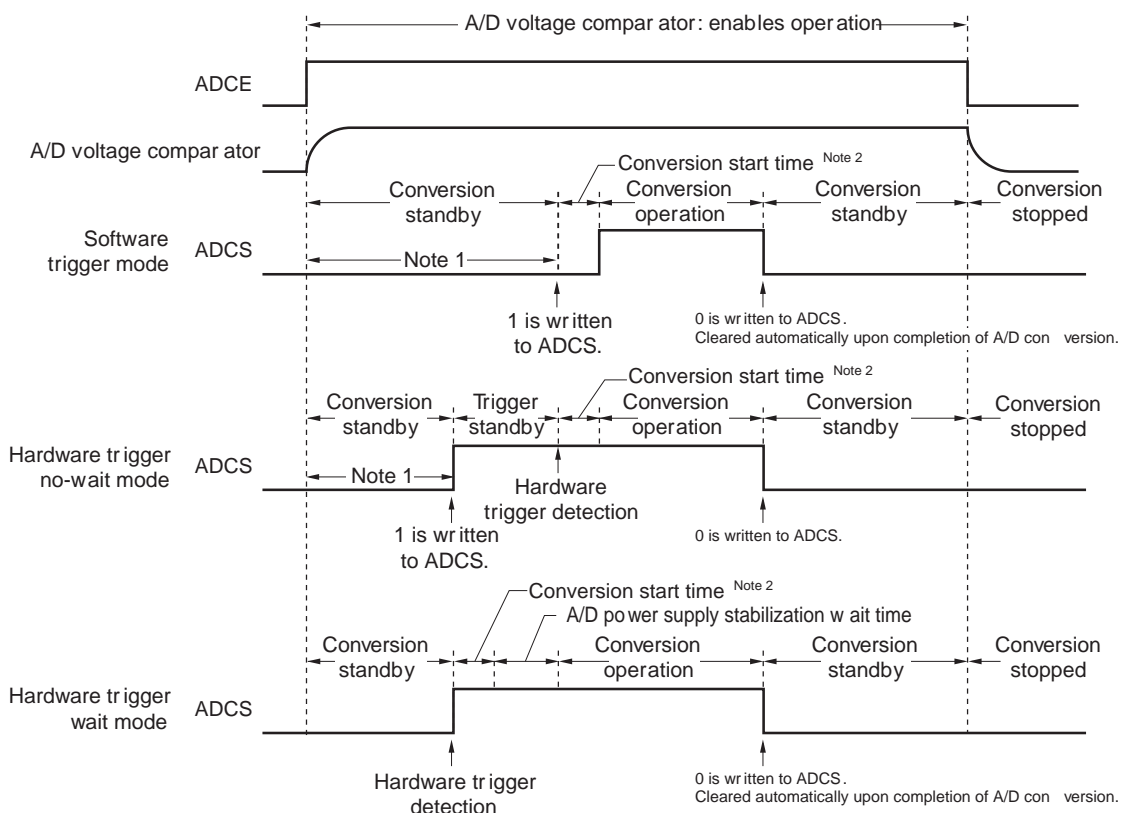
**Table 12-1. Settings of ADCS and ADCE Bits**

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 12-2. Setting and Clearing Conditions for ADCS Bit**

A/D Conversion Mode		Set Conditions	Clear Conditions
Software trigger	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
	One-shot conversion mode		<ul style="list-style-type: none"> <li>When 0 is written to ADCS</li> <li>The bit is automatically cleared to 0 when A/D conversion ends.</li> </ul>
Hardware trigger no-wait mode	Sequential conversion mode		When 0 is written to ADCS
	One-shot conversion mode		When 0 is written to ADCS
Hardware trigger wait mode	Sequential conversion mode	When a hardware trigger is input	When 0 is written to ADCS
	One-shot conversion mode		<ul style="list-style-type: none"> <li>When 0 is written to ADCS</li> <li>The bit is automatically cleared to 0 when A/D conversion ends.</li> </ul>

**Figure 12-4. 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  $\mu$ s or longer to stabilize the internal circuit.
- 2.** The following time (max.) is required until conversion starts:

ADM0			Conversion Clock ( $f_{AD}$ )	Conversion Start Time (Number of $f_{CLK}$ Clock Cycles)	
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, 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.** Rewrite the value of the ADCE bit only when ADCS = 0 (conversion stopped or conversion is on standby).

4. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
- Hardware trigger no wait mode:  $2 f_{\text{CLK}}$  cycles + conversion start time + A/D conversion time
  - Hardware trigger wait mode:  $2 f_{\text{CLK}}$  cycles + conversion start time + A/D power supply stabilization wait time + A/D conversion time

**Remark**  $f_{\text{CLK}}$ : CPU/peripheral hardware clock frequency

**Table 12-3. 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 <sub>AD</sub> )	Number of Conversion Clock Cycles <sup>Note</sup>	Conversion Time	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0					2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V					
									f <sub>CLK</sub> = 1 MHz	f <sub>CLK</sub> = 4 MHz	f <sub>CLK</sub> = 8 MHz	f <sub>CLK</sub> = 16 MHz	f <sub>CLK</sub> = 24 MHz	
0	0	0	0	0	Normal 1	f <sub>CLK</sub> /64	19 f <sub>AD</sub> (number of sampling clock cycles: 7 f <sub>AD</sub> )	1216/f <sub>CLK</sub>	Setting prohibited	Setting prohibited	Setting prohibited	76 μs	50.6667 μs	
0	0	1	f <sub>CLK</sub> /32	608/f <sub>CLK</sub>		76 μs		38 μs				25.3333 μs		
0	1	0	f <sub>CLK</sub> /16	304/f <sub>CLK</sub>		76 μs		38 μs				19 μs	12.6667 μs	
0	1	1	f <sub>CLK</sub> /8	152/f <sub>CLK</sub>		38 μs		19 μs				9.5 μs	6.3333 μs	
1	0	0	f <sub>CLK</sub> /6	114/f <sub>CLK</sub>		28.5 μs		14.25 μs				7.125 μs	4.75 μs	
1	0	1	f <sub>CLK</sub> /5	95/f <sub>CLK</sub>		95 μs		23.75 μs				11.875 μs	5.938 μs	3.9583 μs
1	1	0	f <sub>CLK</sub> /4	76/f <sub>CLK</sub>		76 μs		19 μs				9.5 μs	4.75 μs	3.1667 μs
1	1	1	f <sub>CLK</sub> /2	38/f <sub>CLK</sub>		38 μs		9.5 μs				4.75 μs	2.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f <sub>CLK</sub> /64	17 f <sub>AD</sub> (number of sampling clock cycles: 5 f <sub>AD</sub> )	1088/f <sub>CLK</sub>	Setting prohibited	Setting prohibited	Setting prohibited	68 μs	45.3333 μs	
0	0	1	f <sub>CLK</sub> /32	544/f <sub>CLK</sub>		68 μs		34 μs				22.6667 μs		
0	1	0	f <sub>CLK</sub> /16	272/f <sub>CLK</sub>		68 μs		34 μs				17 μs	11.3333 μs	
0	1	1	f <sub>CLK</sub> /8	136/f <sub>CLK</sub>		34 μs		17 μs				8.5 μs	5.6667 μs	
1	0	0	f <sub>CLK</sub> /6	102/f <sub>CLK</sub>		25.5 μs		12.75 μs				6.375 μs	4.25 μs	
1	0	1	f <sub>CLK</sub> /5	85/f <sub>CLK</sub>		85 μs		21.25 μs				10.625 μs	5.3125 μs	3.5417 μs
1	1	0	f <sub>CLK</sub> /4	68/f <sub>CLK</sub>		68 μs		17 μs				8.5 μs	4.25 μs	2.8333 μs
1	1	1	f <sub>CLK</sub> /2	34/f <sub>CLK</sub>		34 μs		8.5 μs				4.25 μs	2.125 μs	Setting prohibited

**Note** 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<sub>AD</sub>).

- Cautions**
1. The A/D conversion time must also be within the relevant range of conversion times (t<sub>CONV</sub>) described in 32.6.1 A/D converter characteristics or 33.6.1 A/D converter characteristics.
  2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, make sure that conversion has 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<sub>CLK</sub>: CPU/peripheral hardware clock frequency

**Table 12-3. A/D Conversion Time Selection (2/4)**

(2) When there is no A/D power supply stabilization wait time  
 Low-voltage mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f <sub>AD</sub> )	Number of Conversion Clock Cycles <sup>Note 4</sup>	Conversion Time	Conversion Time Selection				
FR2	FR1	FR0	LV1	LV0					1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V				
									f <sub>CLK</sub> = 1 MHz	f <sub>CLK</sub> = 4 MHz	f <sub>CLK</sub> = 8 MHz	f <sub>CLK</sub> = 16 MHz	f <sub>CLK</sub> = 24 MHz
0	0	0	1	0	Low-voltage 1	f <sub>CLK</sub> /64	19 f <sub>AD</sub> (number of sampling clock cycles: 7 f <sub>AD</sub> )	1216/f <sub>CLK</sub>	Setting prohibited	Setting prohibited	Setting prohibited	76 μs	50.6667 μs
0	0	1	f <sub>CLK</sub> /32	608/f <sub>CLK</sub>		76 μs		38 μs				25.3333 μs	
0	1	0	f <sub>CLK</sub> /16	304/f <sub>CLK</sub>		38 μs		19 μs				12.6667 μs	
0	1	1	f <sub>CLK</sub> /8	152/f <sub>CLK</sub>		28.5 μs		14.25 μs				6.3333 μs	
1	0	0	f <sub>CLK</sub> /6	114/f <sub>CLK</sub>		23.75 μs		11.875 μs				4.75 μs	
1	0	1	f <sub>CLK</sub> /5	95/f <sub>CLK</sub>		19 μs		9.5 μs				3.9587 μs	
1	1	0	f <sub>CLK</sub> /4	76/f <sub>CLK</sub>		14.25 μs		7.125 μs				3.1667 μs	
1	1	1	f <sub>CLK</sub> /2	38/f <sub>CLK</sub>		9.5 μs		4.75 μs				Setting prohibited	
0	0	0	1	1	Low-voltage 2	f <sub>CLK</sub> /64	17 f <sub>AD</sub> (number of sampling clock cycles: 5 f <sub>AD</sub> )	1088/f <sub>CLK</sub>	Setting prohibited	Setting prohibited	Setting prohibited	68 μs	45.3333 μs
0	0	1	f <sub>CLK</sub> /32	544/f <sub>CLK</sub>		68 μs		34 μs				22.6667 μs	
0	1	0	f <sub>CLK</sub> /16	272/f <sub>CLK</sub>		34 μs		17 μs				11.3333 μs	
0	1	1	f <sub>CLK</sub> /8	136/f <sub>CLK</sub>		25.5 μs		12.75 μs				5.6667 μs	
1	0	0	f <sub>CLK</sub> /6	102/f <sub>CLK</sub>		21.25 μs		10.625 μs				4.25 μs	
1	0	1	f <sub>CLK</sub> /5	85/f <sub>CLK</sub>		17 μs		8.5 μs				3.5417 μs	
1	1	0	f <sub>CLK</sub> /4	68/f <sub>CLK</sub>		14.25 μs		7.125 μs				2.8333 μs	
1	1	1	f <sub>CLK</sub> /2	34/f <sub>CLK</sub>		9.5 μs		4.75 μs				Setting prohibited	

- Notes**
- 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V
  - 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V
  - 2.7 V ≤ V<sub>DD</sub> ≤ 5.5 V
  - 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<sub>AD</sub>).

- Cautions**
- The A/D conversion time must also be within the relevant range of conversion times (t<sub>CONV</sub>) described in 32.6.1 A/D converter characteristics or 33.6.1 A/D converter characteristics.
  - When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, make sure that conversion has stopped (ADCS = 0, ADCE = 0).
  - 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<sub>CLK</sub>: CPU/peripheral hardware clock frequency

**Table 12-3. 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<sup>Note 1</sup>)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f <sub>AD</sub> )	Number of Stabilization Wait Cycles	Number of Conversion Clock Cycles <sup>Note 2</sup>	Stabilization Wait Cycles + Conversion Time	Conversion Time Selection					
FR 2	FR 1	FR 0	LV 1	LV 0						2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V					
										f <sub>CLK</sub> = 1 MHz	f <sub>CLK</sub> = 4 MHz	f <sub>CLK</sub> = 8 MHz	f <sub>CLK</sub> = 16 MHz	f <sub>CLK</sub> = 24 MHz	
0	0	0	0	0	Normal 1	f <sub>CLK</sub> /64	8 f <sub>AD</sub>	19 f <sub>AD</sub> (number of sampling clock cycles: 7 f <sub>AD</sub> )	1728/f <sub>CLK</sub>  864/f <sub>CLK</sub>  432/f <sub>CLK</sub>  216/f <sub>CLK</sub>  162/f <sub>CLK</sub>  135/f <sub>CLK</sub>  108/f <sub>CLK</sub>  54/f <sub>CLK</sub>	Setting prohibited	Setting prohibited	Setting prohibited	108 μs	72 μs	
0	0	1	f <sub>CLK</sub> /32	108 μs		54 μs							36 μs		
0	1	0	f <sub>CLK</sub> /16	108 μs		54 μs							27 μs	18 μs	
0	1	1	f <sub>CLK</sub> /8	54 μs		27 μs							13.5 μs	9 μs	
1	0	0	f <sub>CLK</sub> /6	40.5 μs		20.25 μs							10.125 μs	6.75 μs	
1	0	1	f <sub>CLK</sub> /5	135 μs		33.75 μs							16.875 μs	8.4375 μs	5.625 μs
1	1	0	f <sub>CLK</sub> /4	108 μs		27 μs							13.5 μs	6.75 μs	4.5 μs
1	1	1	f <sub>CLK</sub> /2	54 μs		13.5 μs							6.75 μs	3.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f <sub>CLK</sub> /64	8 f <sub>AD</sub>	17 f <sub>AD</sub> (number of sampling clock cycles: 5 f <sub>AD</sub> )	1600/f <sub>CLK</sub>  800/f <sub>CLK</sub>  400/f <sub>CLK</sub>  200/f <sub>CLK</sub>  150/f <sub>CLK</sub>  125/f <sub>CLK</sub>  100/f <sub>CLK</sub>  50/f <sub>CLK</sub>	Setting prohibited	Setting prohibited	Setting prohibited	100 μs	66.6667 μs	
0	0	1	f <sub>CLK</sub> /32	100 μs		50 μs							33.3333 μs		
0	1	0	f <sub>CLK</sub> /16	100 μs		50 μs							25 μs	16.6667 μs	
0	1	1	f <sub>CLK</sub> /8	50 μs		25 μs							12.5 μs	8.3333 μs	
1	0	0	f <sub>CLK</sub> /6	37.5 μs		18.75 μs							9.375 μs	6.25 μs	
1	0	1	f <sub>CLK</sub> /5	125 μs		31.25 μs							15.625 μs	7.8125 μs	5.2083 μs
1	1	0	f <sub>CLK</sub> /4	100 μs		25 μs							12.5 μs	6.25 μs	4.1667 μs
1	1	1	f <sub>CLK</sub> /2	50 μs		12.5 μs							6.25 μs	3.125 μs	Setting prohibited

- Notes**
- For the second and subsequent conversion in sequential conversion 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 12-3 (1/4)**).
  - 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<sub>AD</sub>).

- Cautions**
- The A/D conversion time must also be within the relevant range of conversion times (t<sub>CONV</sub>) described in 32.6.1 A/D converter characteristics or 33.6.1 A/D converter characteristics.
  - When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, make sure that conversion has stopped (ADCS = 0, ADCE = 0).
  - 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.
  - When hardware trigger wait mode, specify the conversion time, including the stabilization wait time from the hardware trigger detection.

**Remark** f<sub>CLK</sub>: CPU/peripheral hardware clock frequency

**Table 12-3. A/D Conversion Time Selection (4/4)**

(4) When there is A/D power supply stabilization wait time  
 Low-voltage mode 1, 2 (hardware trigger wait mode<sup>Note 1</sup>)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f <sub>AD</sub> )	Number of Stabilization Wait Cycles	Number of Conversion Clock Cycles <sup>Note 5</sup>	Stabilization Wait Cycles + Conversion Time	Conversion Time Selection				
FR 2	FR 1	FR 0	LV 1	LV 0						1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V		Note 2	Note 3	Note 4
									f <sub>CLK</sub> = 1 MHz	f <sub>CLK</sub> = 4 MHz	f <sub>CLK</sub> = 8 MHz	f <sub>CLK</sub> = 16 MHz	f <sub>CLK</sub> = 24 MHz	
0	0	0	0	0	Low voltage 1	f <sub>CLK</sub> /64	2 f <sub>AD</sub>	19 f <sub>AD</sub> (number of sampling clock cycles: 7 f <sub>AD</sub> )	1344/f <sub>CLK</sub>	Setting prohibited	Setting prohibited	Setting prohibited	84 μs	56 μs
0	0	1	f <sub>CLK</sub> /32	672/f <sub>CLK</sub>		84 μs				42 μs	28 μs			
0	1	0	f <sub>CLK</sub> /16	336/f <sub>CLK</sub>		84 μs				42 μs	21 μs	14 μs		
0	1	1	f <sub>CLK</sub> /8	168/f <sub>CLK</sub>		42 μs				21 μs	10.5 μs	7 μs		
1	0	0	f <sub>CLK</sub> /6	126/f <sub>CLK</sub>		31.25 μs				15.75 μs	7.875 μs	5.25 μs		
1	0	1	f <sub>CLK</sub> /5	105/f <sub>CLK</sub>		105 μs				26.25 μs	13.125 μs	6.5625 μs	4.375 μs	
1	1	0	f <sub>CLK</sub> /4	84/f <sub>CLK</sub>		84 μs				21 μs	10.5 μs	5.25 μs	3.5 μs	
1	1	1	f <sub>CLK</sub> /2	42/f <sub>CLK</sub>		42 μs				10.5 μs	5.25 μs	2.625 μs	Setting prohibited	
0	0	0	0	1	Low voltage 2	f <sub>CLK</sub> /64	2 f <sub>AD</sub>	17 f <sub>AD</sub> (number of sampling clock cycles: 5 f <sub>AD</sub> )	1216/f <sub>CLK</sub>	Setting prohibited	Setting prohibited	Setting prohibited	76 μs	50.6667 μs
0	0	1	f <sub>CLK</sub> /32	608/f <sub>CLK</sub>		76 μs				38 μs	25.3333 μs			
0	1	0	f <sub>CLK</sub> /16	304/f <sub>CLK</sub>		76 μs				38 μs	19 μs	12.6667 μs		
0	1	1	f <sub>CLK</sub> /8	152/f <sub>CLK</sub>		38 μs				19 μs	9.5 μs	6.3333 μs		
1	0	0	f <sub>CLK</sub> /6	114/f <sub>CLK</sub>		28.5 μs				14.25 μs	7.125 μs	4.75 μs		
1	0	1	f <sub>CLK</sub> /5	96/f <sub>CLK</sub>		96 μs				23.75 μs	12 μs	5.938 μs	4.0 μs	
1	1	0	f <sub>CLK</sub> /4	76/f <sub>CLK</sub>		76 μs				19 μs	9.5 μs	4.75 μs	3.1667 μs	
1	1	1	f <sub>CLK</sub> /2	38/f <sub>CLK</sub>		38 μs				9.5 μs	4.75 μs	2.375 μs	Setting prohibited	

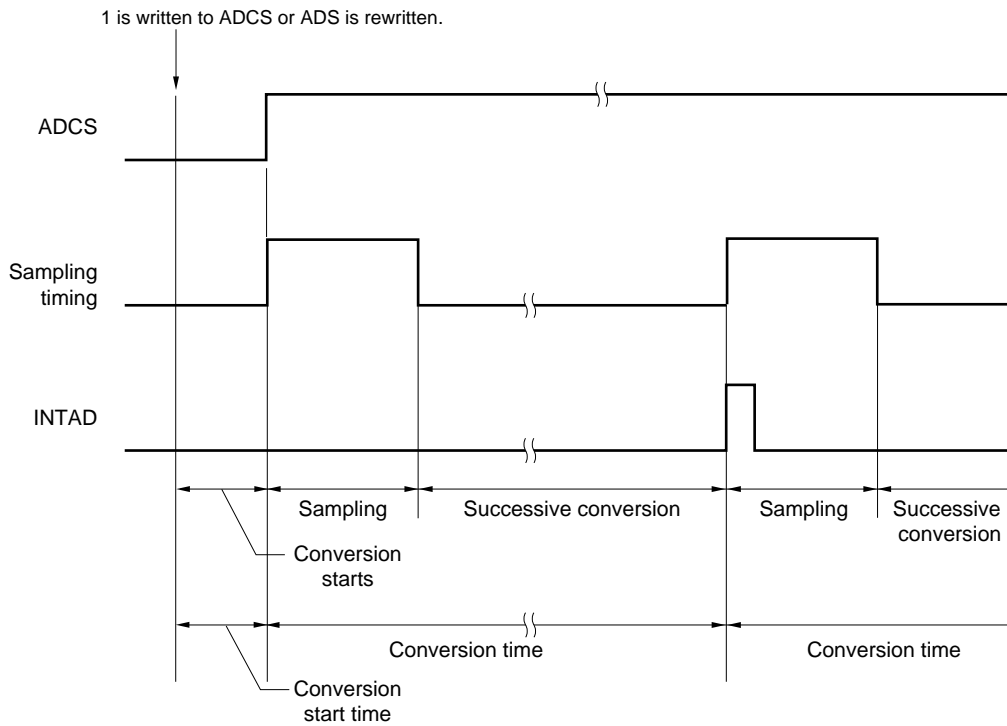
- Notes**
- For the second and subsequent conversion in sequential conversion mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 12-3 (2/4)**).
  - 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V
  - 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V
  - 2.7 V ≤ V<sub>DD</sub> ≤ 5.5 V
  - 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<sub>AD</sub>).

- Cautions**
- The A/D conversion time must also be within the relevant range of conversion times (t<sub>CONV</sub>) described in 32.6.1 A/D converter characteristics or 33.6.1 A/D converter characteristics.
  - When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, make sure that conversion has stopped (ADCS = 0, ADCE = 0).
  - 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.
  - When hardware trigger wait mode, specify the conversion time, including the stabilization wait time from the hardware trigger detection.

**Remark** f<sub>CLK</sub>: CPU/peripheral hardware clock frequency



Figure 12-5. A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



**12.3.3 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 12-6. 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 01 count or capture interrupt signal (INTTM01)
0	1	Setting prohibited
1	0	Real-time clock 2 interrupt signal (INTRTC)
1	1	12-bit interval timer interrupt signal (INTIT)

- Cautions**
1. Only rewrite the value of the ADM1 register while conversion operation is stopped (which is indicated by the ADCS bit of A/D converter mode register 0 (ADM0) being 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<sub>CLK</sub> cycles + conversion start time + A/D conversion time  
 Hardware trigger wait mode: 2 f<sub>CLK</sub> cycles + 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<sub>CLK</sub> cycles after the first INTRTC or INTIT is input.

- Remarks**
1. ×: don't care
  2. f<sub>CLK</sub>: CPU/peripheral hardware clock frequency

**12.3.4 A/D converter mode register 2 (ADM2)**

This register is used to select the A/D converter reference voltage, 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 12-7. 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 of the A/D converter
0	0	Supplied from V <sub>DD</sub>
0	1	Supplied from P21/AV <sub>REFP</sub> /ANI0
1	0	Supplied from the internal reference voltage (1.45 V) <sup>Note</sup>
1	1	Setting prohibited

- When ADREFP1 or ADREFP0 bit is rewritten, this must be configured in accordance with the following procedures.
  - Set ADCE = 0
  - Change the values of ADREFP1 and ADREFP0
  - Reference voltage stabilization wait time (A)
  - Set ADCE = 1
  - 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.  
 After (5) stabilization time, start the A/D conversion.
- When ADREFP1 and ADREFP0 are set to 1 and 0, respectively, A/D conversion cannot be performed on the temperature sensor output voltage and internal reference voltage (1.45 V).  
Be sure to perform A/D conversion while ADISS = 0.

ADREFM	Selection of the – side reference voltage of the A/D converter
0	Supplied from V <sub>SS</sub>
1	Supplied from P20/AV <sub>REFM</sub> /ANI1

**Note** This setting can be used only in HS (high-speed main) mode.

- Cautions**
- Rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).
  - 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. When the internal reference voltage is selected (ADREFP1, ADREFP0 = 1, 0), the A/D converter reference voltage current (I<sub>ADREF</sub>) indicated in 32.4.2 Supply current characteristics or 33.4.2 Supply current characteristics will be added.
  - When using AV<sub>REFP</sub> and AV<sub>REFM</sub>, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.

**Figure 12-7. 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

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signal (INTAD) is output when the ADLL register $\leq$ the ADCR register $\leq$ the ADUL register (AREA 1).
1	The interrupt signal (INTAD) is output when the ADCR register $<$ the ADLL register (AREA 2) or the ADUL register $<$ the ADCR register (AREA 3).
Figure 12-8 shows the generation range of the interrupt signal (INTAD) for AREA 1 to AREA 3.	

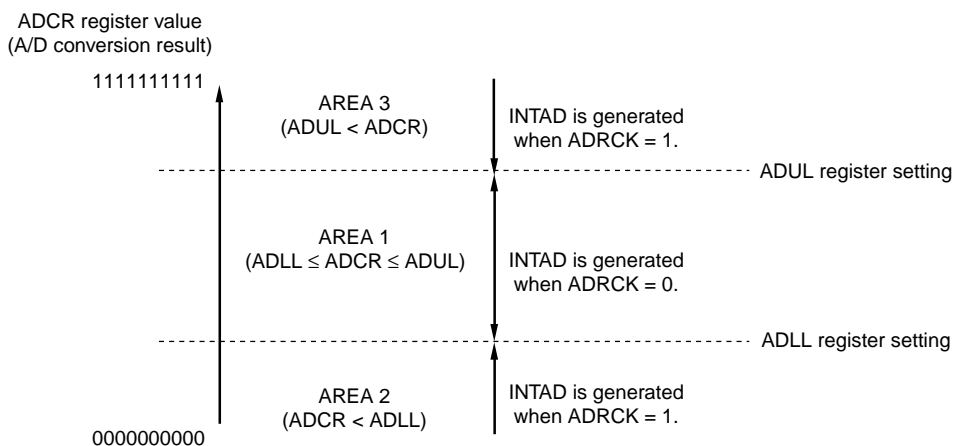
AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.
<p>When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode).</p> <ul style="list-style-type: none"> <li>The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (<math>f_{CLK}</math>). If any other clock is selected, specifying this mode is prohibited.</li> <li>Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited.</li> <li>Using the SNOOZE mode function in the sequential conversion mode is prohibited.</li> <li>When using the SNOOZE mode function, specify a hardware trigger interval of at least "shift time to SNOOZE mode<sup>Note</sup> + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 2 <math>f_{CLK}</math> clock"</li> <li>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.</li> </ul> <p>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.</p>	

ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

**Note** See "From STOP to SNOOZE" in 21.3.3 SNOOZE mode.

**Caution** Only rewrite the value of the ADM2 register while conversion operation is stopped (ADCS = 0, ADCE = 0).

**Figure 12-8. 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.

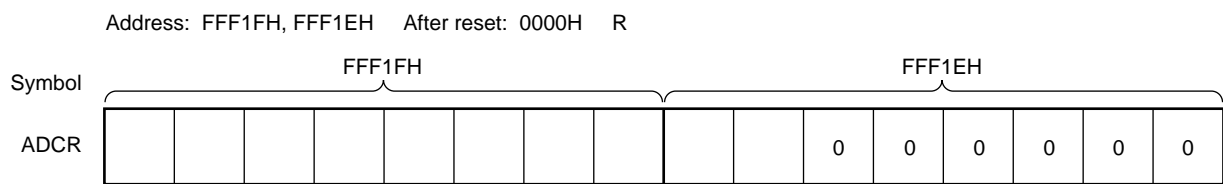
**12.3.5 10-bit A/D conversion result register (ADCR)**

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 <sup>Note</sup>.

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 12-8), the result is not stored.

**Figure 12-9. Format of 10-bit A/D Conversion Result Register (ADCR)**



- 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 (bits 7 and 6 of the ADCR register).
  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.

### 12.3.6 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 <sup>Note</sup>. 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 12-8**), the result is not stored.

**Figure 12-10. 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.

### 12.3.7 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 12-11. 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

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P21/ANI0/AV <sub>REFP</sub> pin
0	0	0	0	0	1	ANI1	P20/ANI1/AV <sub>REFM</sub> pin
0	1	0	0	0	0	ANI16	P22/ANI16 pin
0	1	0	0	0	1	ANI17	P23/ANI17 pin
0	1	0	0	1	0	ANI18	P24/ANI18 pin
0	1	0	0	1	1	ANI19	P25/ANI19 pin
0	1	0	1	0	0	ANI20	P26/ANI20 pin
0	1	0	1	0	1	ANI21	P27/ANI21 pin
0	1	0	1	1	0	ANI22	P10/ANI22 pin
0	1	0	1	1	1	ANI23	P11/ANI23 pin
0	1	1	0	0	0	ANI24	P12/ANI24 pin
0	1	1	0	0	1	ANI25	P13/ANI25 pin
1	0	0	0	0	0	–	Temperature sensor output <sup>Note</sup>
1	0	0	0	0	1	–	Internal reference voltage (1.45 V) <sup>Note</sup>
Other than above						Setting prohibited	

**Note** 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. Set a channel to be set the analog input by ADPC and PMC registers in the input mode by using port mode registers 1, 2 (PM1, PM2).
  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 registers 1, 2 (PMC1, PMC2) as digital I/O by the ADS register.
  5. Only rewrite the value of the ADISS bit while conversion operation is stopped (which is indicated by the ADCE bit of A/D voltage cooperator mode register 0 (ADM0) being 0).
  6. If using AV<sub>REFP</sub> as the + side reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.
  7. If using AV<sub>REFM</sub> as the – side reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.
  8. If ADISS is set to 1, the internal reference voltage (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 12.7.4 Setup when using temperature sensor (example for software trigger mode and one-shot conversion mode).

- Cautions**
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<sub>ADREF</sub>) indicated in 32.3.2 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.
  10. Ignore the conversion result if the corresponding ANI pin does not exist in the product used.

**Remark** x: don't care

**12.3.8 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 12-8**).

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 12-12. 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

**12.3.9 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 12-8**).

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

Reset signal generation clears this register to 00H.

**Figure 12-13. 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 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 ADLL register.
  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.



**12.3.10 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 (ANlxx), the temperature sensor output voltage, or the internal reference voltage (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 12-14. 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	ANlxx/temperature sensor output voltage <sup>Note</sup> /internal reference voltage (1.45 V) <sup>Note</sup> (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 above		Setting prohibited

**Note** The temperature sensor output voltage and internal reference voltage (1.45 V) can be selected only in the HS (high-speed main) mode.

**12.3.11 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 and ANI1 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 ANI16 to ANI25 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.

## 12.4 A/D Converter Conversion Operations

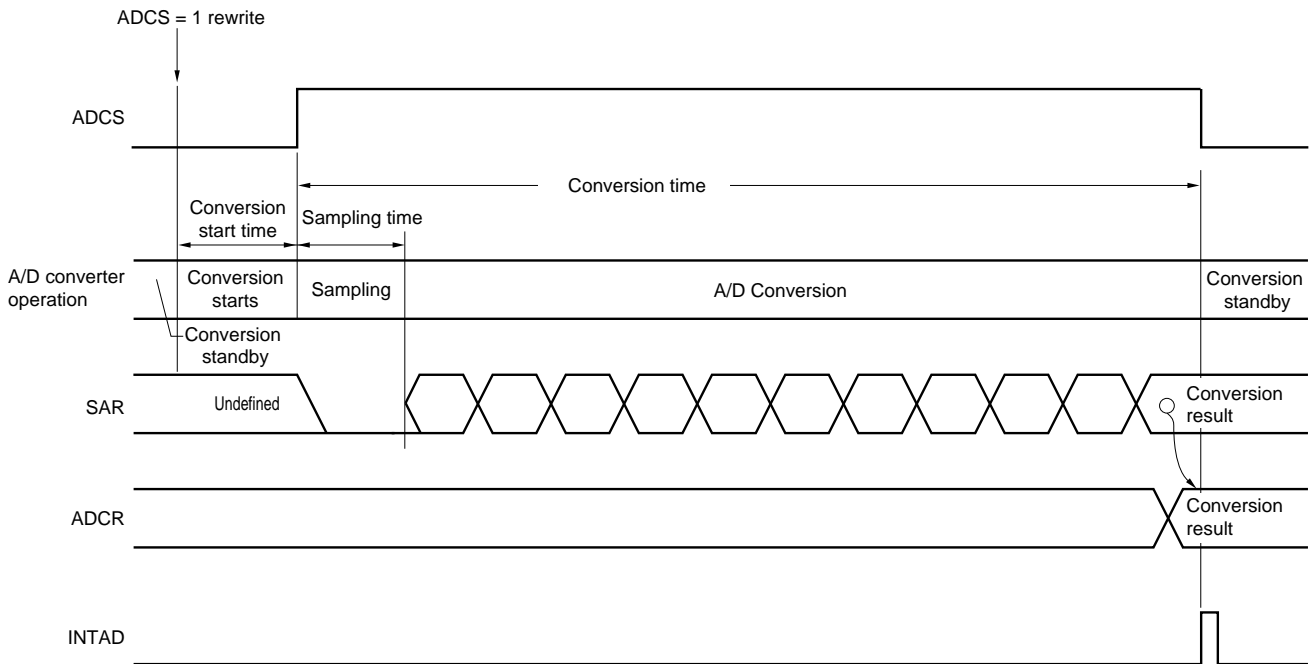
The A/D converter conversion operations are described below.

- <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  $\geq$  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<sup>Note 1</sup>. At the same time, the A/D conversion end interrupt request (INTAD) can also be generated<sup>Note 1</sup>.
- <8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0<sup>Note 2</sup>.  
To stop the A/D converter, clear the ADCS bit to 0.

- Notes**
1. If the A/D conversion result is outside the A/D conversion result range specified by the ADRCK bit and the ADUL and ADLL registers (see **Figure 12-8**), the A/D conversion result interrupt request signal is not generated and no A/D conversion results are stored in the ADCR and ADCRH registers.
  2. 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 12-15. 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.

When the value of the analog input channel specification register (ADS) is rewritten or overwritten during conversion, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input newly specified in the ADS register. The partially converted data is discarded.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

### 12.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0, ANI1, ANI16 to ANI25) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = \text{INT} \left( \frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5 \right)$$

$$ADCR = SAR \times 64$$

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

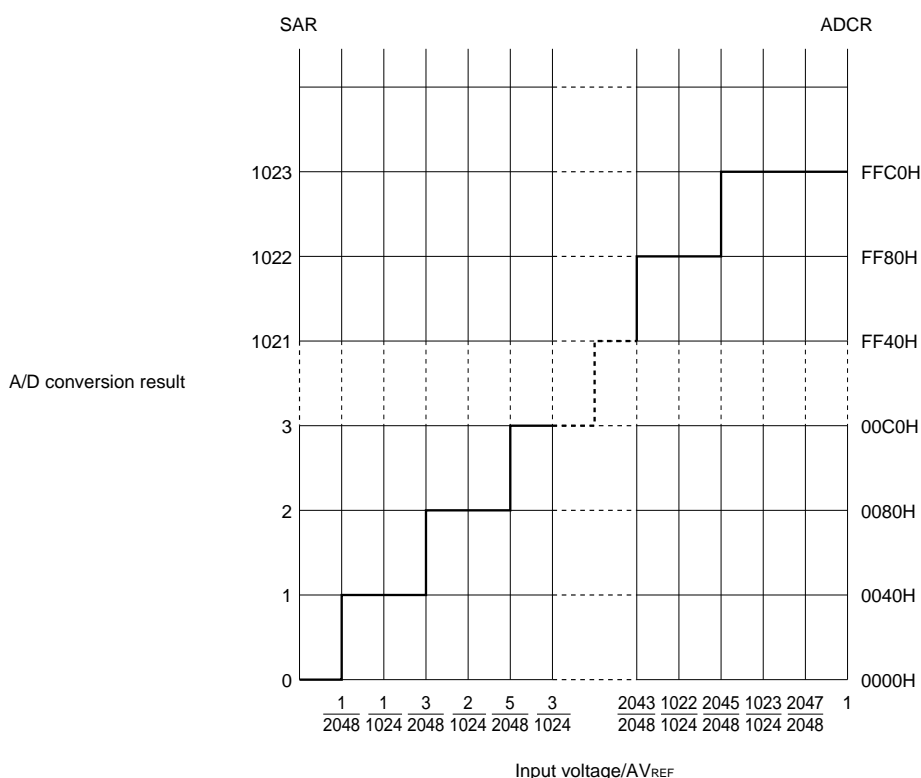
$AV_{REF}$ :  $AV_{REF}$  pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 12-16 shows the relationship between the analog input voltage and the A/D conversion result.

**Figure 12-16. 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  $V_{DD}$ .

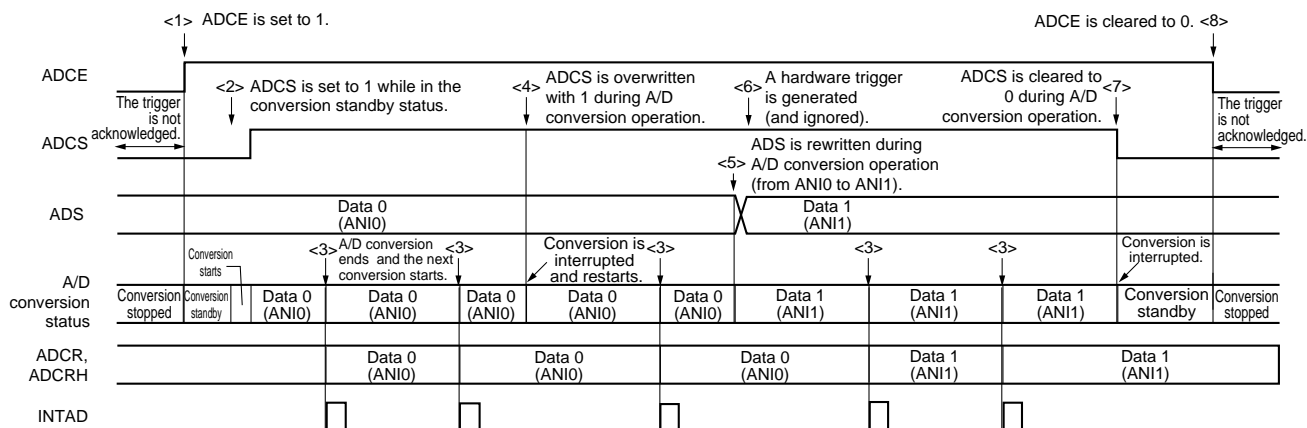
### 12.6 A/D Converter Operation Modes

The operation of each A/D converter mode is described below. In addition, the procedure for specifying each mode is described in 12.7 A/D Converter Setup Flowchart.

#### 12.6.1 Software trigger 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  $\mu$ s), 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.

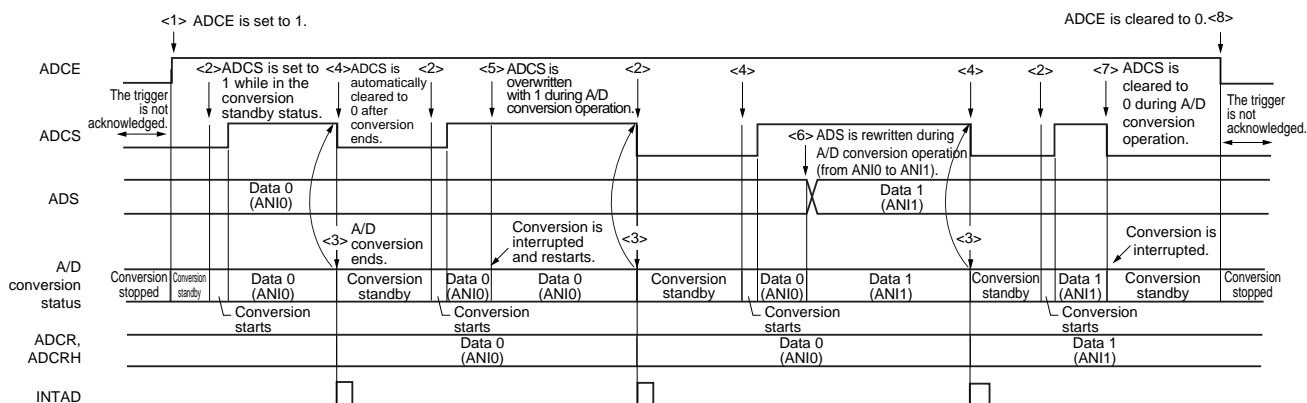
Figure 12-17. Example of Software Trigger Mode (Sequential Conversion Mode) Operation Timing



12.6.2 Software trigger 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  $\mu$ s), 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.

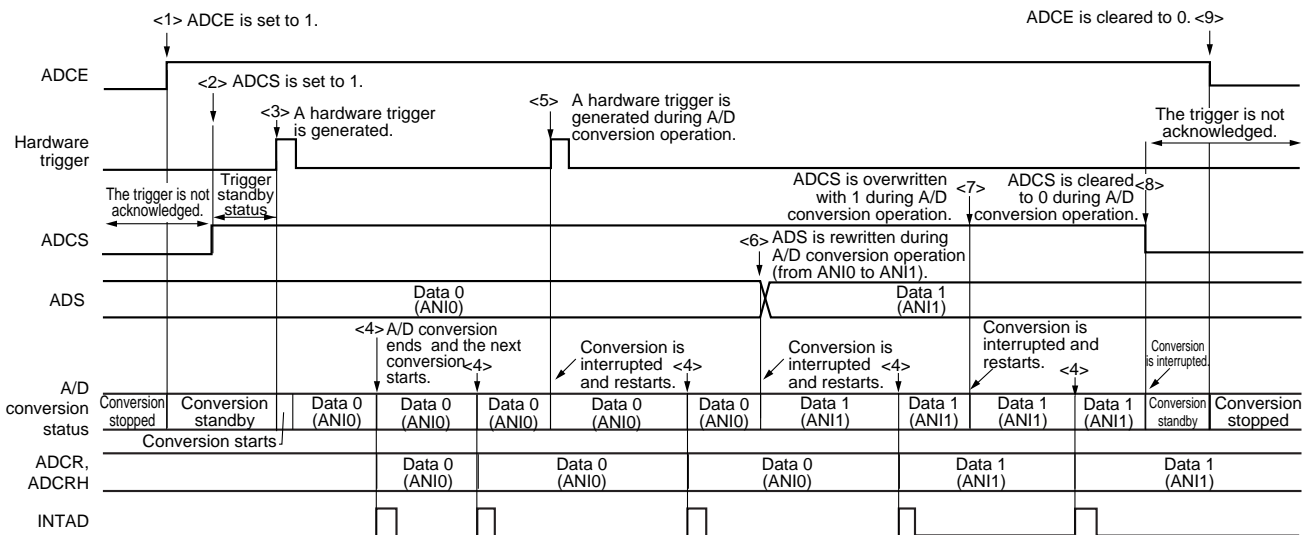
Figure 12-18. Example of Software Trigger Mode (One-Shot Conversion Mode) Operation Timing



12.6.3 Hardware trigger no-wait 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  $\mu$ s), 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.

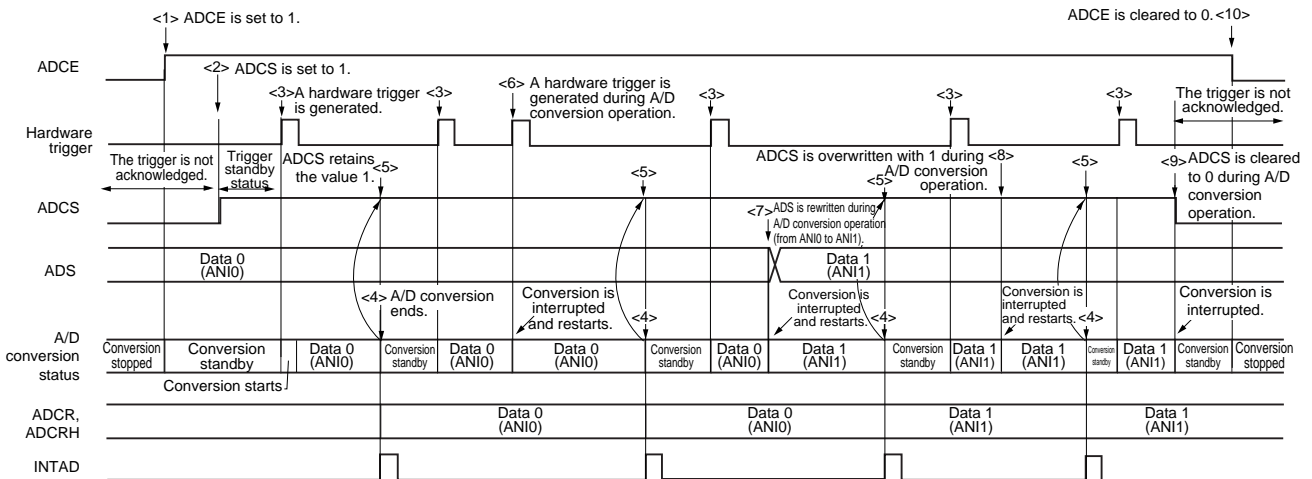
Figure 12-19. Example of Hardware Trigger No-Wait Mode (Sequential Conversion Mode) Operation Timing



12.6.4 Hardware trigger no-wait 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  $\mu$ s), 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 12-20. Example of Hardware Trigger No-Wait Mode (One-Shot Conversion Mode) Operation Timing

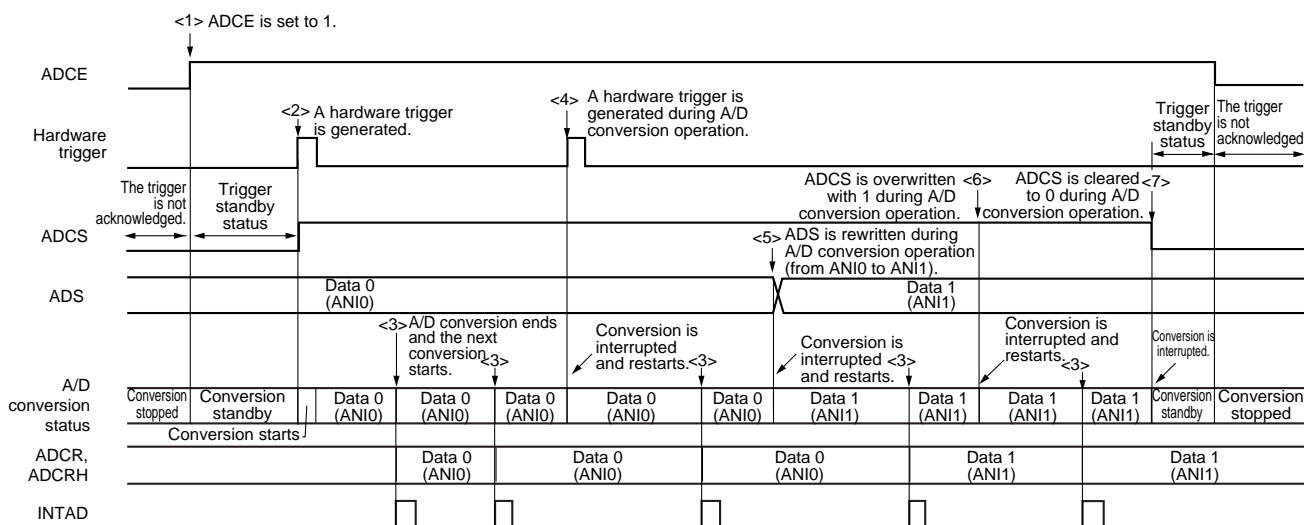




12.6.5 Hardware trigger wait mode (sequential conversion mode)

- <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.

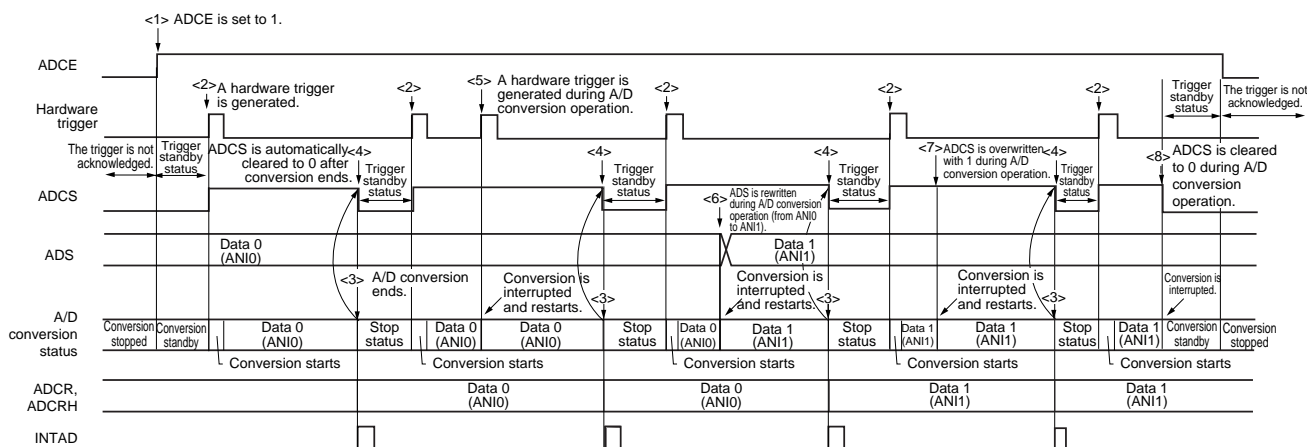
Figure 12-21. Example of Hardware Trigger Wait Mode (Sequential Conversion Mode) Operation Timing



12.6.6 Hardware trigger wait 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 12-22. Example of Hardware Trigger Wait Mode (One-Shot Conversion Mode) Operation Timing

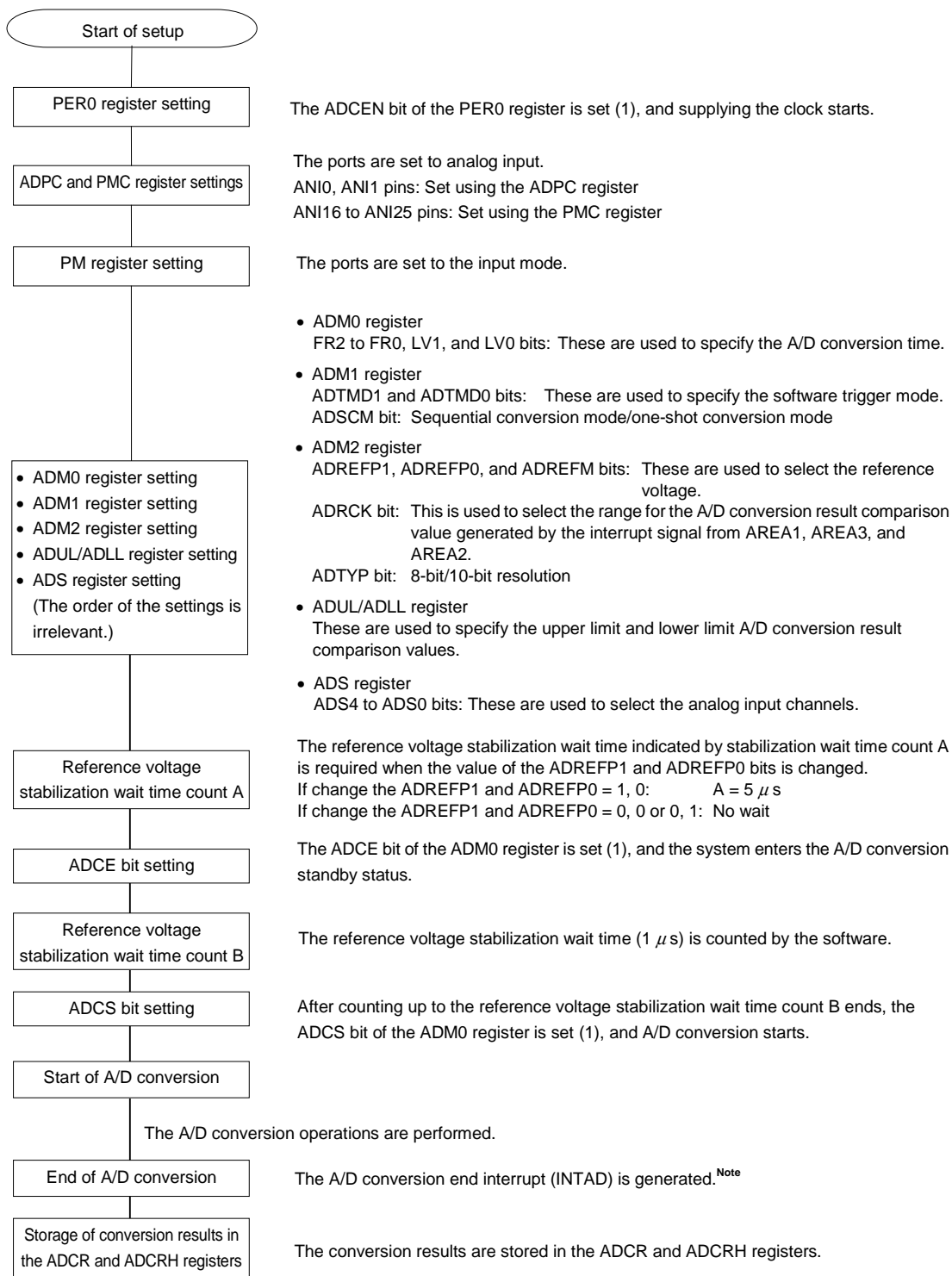


## 12.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

### 12.7.1 Setting up software trigger mode

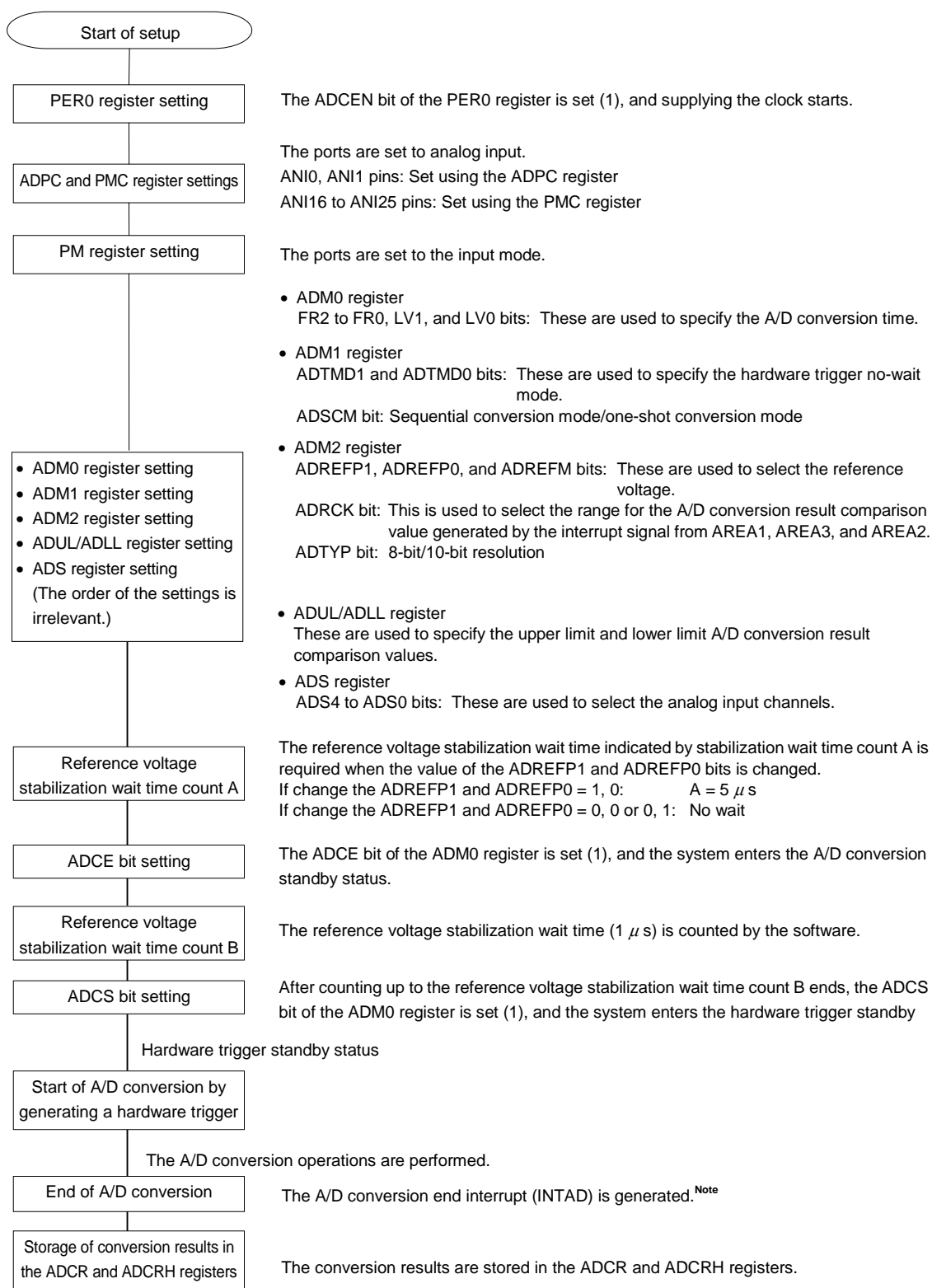
Figure 12-23. Setting up Software Trigger Mode



**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, ADCRH registers.

12.7.2 Setting up hardware trigger no-wait mode

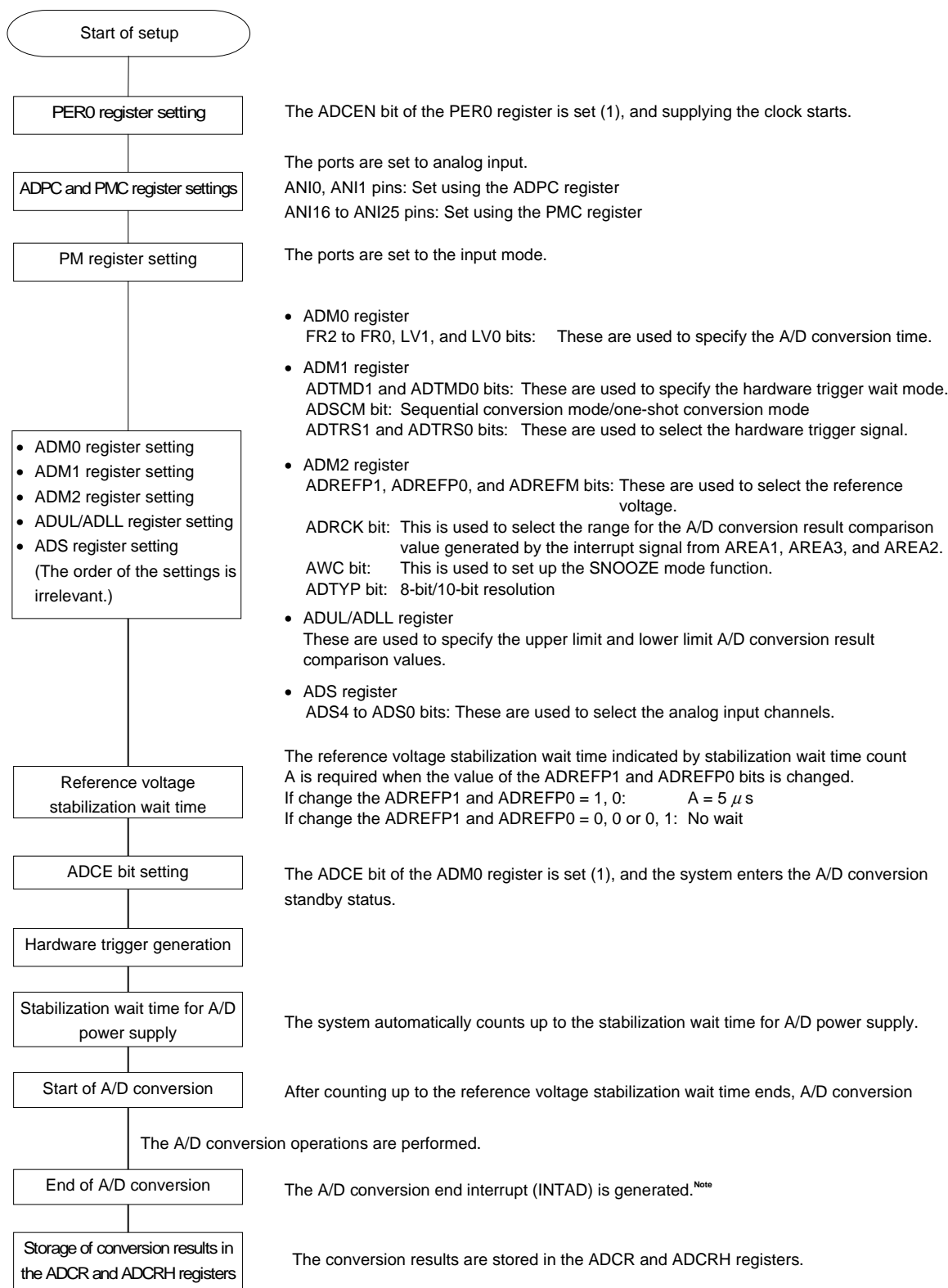
Figure 12-24. Setting up Hardware Trigger No-Wait Mode



**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.

12.7.3 Setting up hardware trigger wait mode

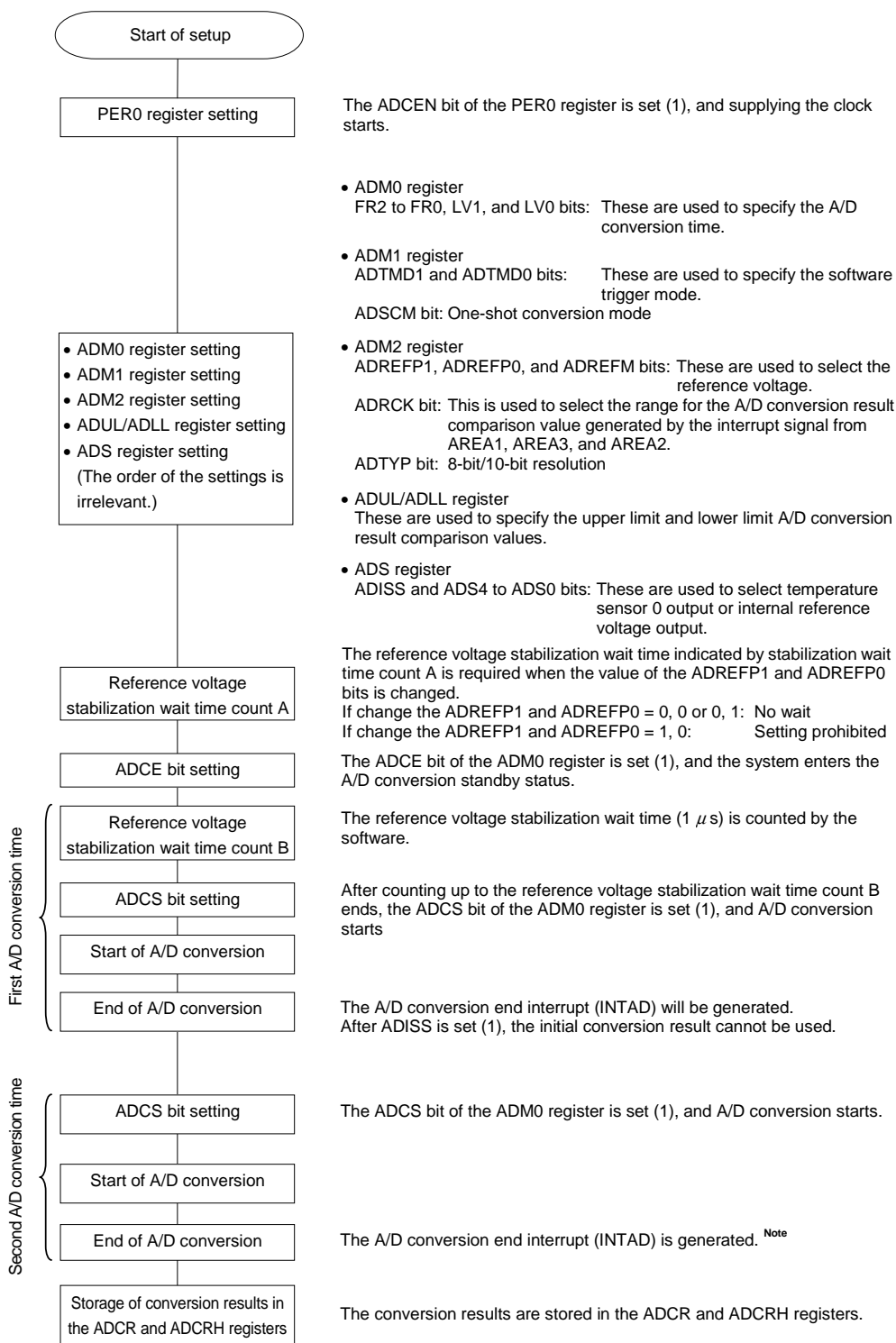
Figure 12-25. Setting up Hardware Trigger Wait Mode



**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.

12.7.4 Setup when using temperature sensor (example for software trigger mode and one-shot conversion mode)

Figure 12-26. Setup When Using Temperature Sensor



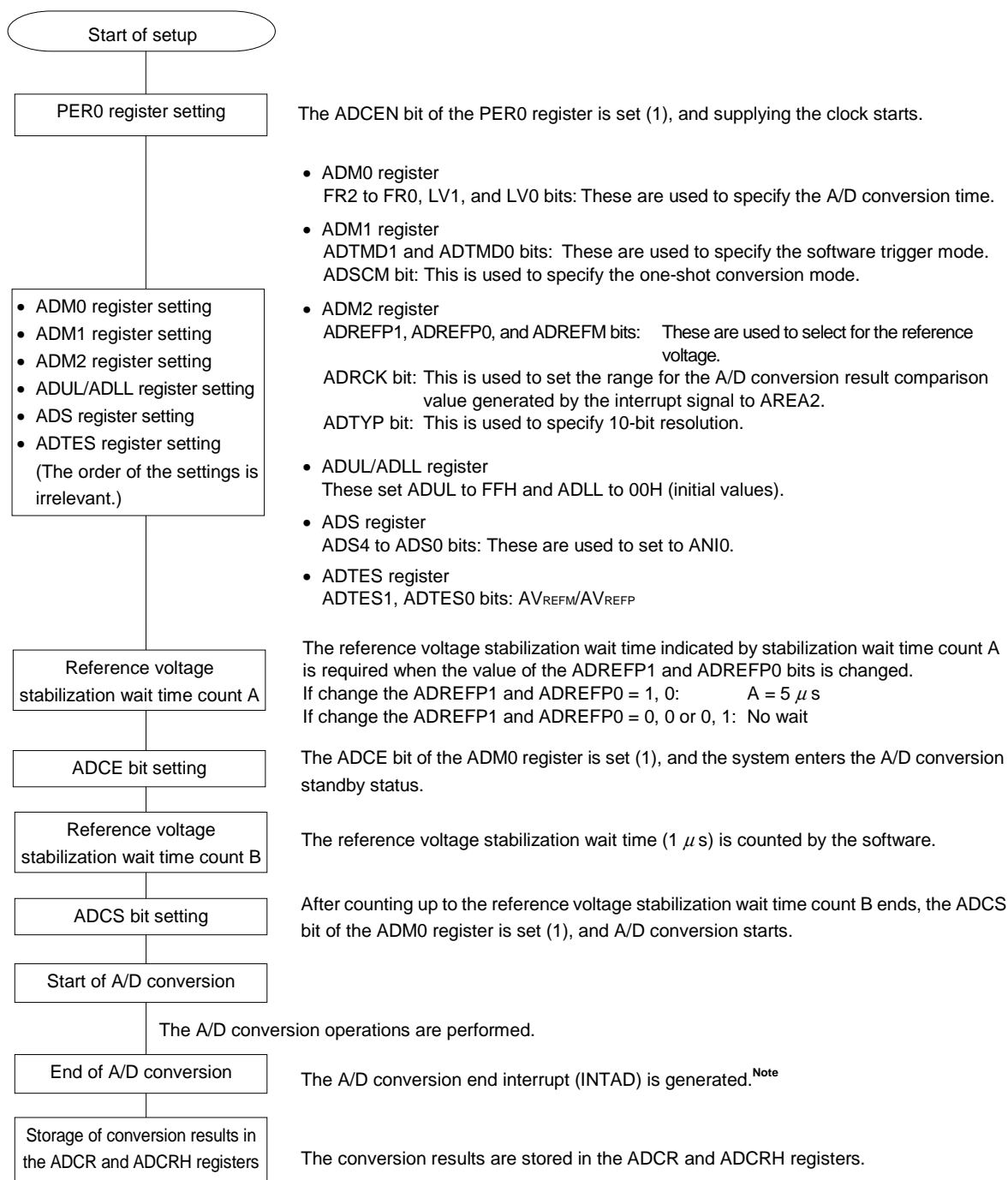
**Note** Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

**Caution** This setting can be used only in HS (high-speed main) mode.

12.7.5 Setting up test mode

<R>

Figure 12-27. Setting up Test 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.

**Caution** For the procedure for testing the A/D converter, see 25.3.8 A/D test function.

## 12.8 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.

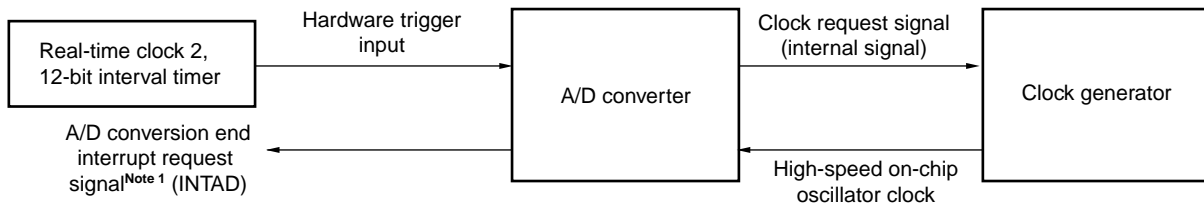
If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time in SNOOZE mode. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

In the SNOOZE mode, only the following conversion modes can be used:

- Hardware trigger wait 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<sub>CLK</sub>.

Figure 12-28. 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 **12.7.3 Setting up hardware trigger wait mode**<sup>Note 2</sup>). 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 generated<sup>Note 1</sup>.

**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).

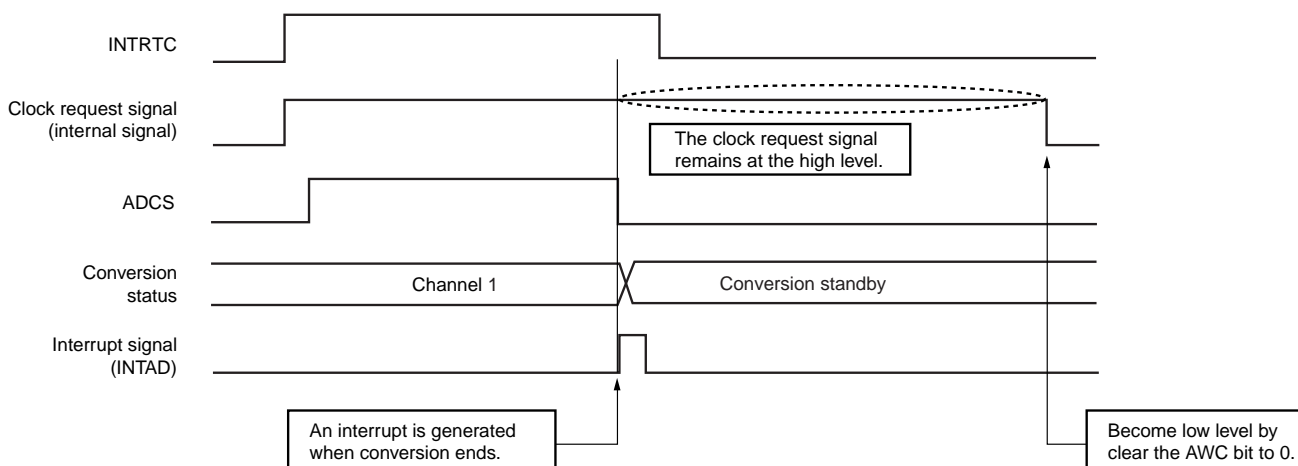


**(1) If an interrupt is generated after A/D conversion ends**

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.

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.

**Figure 12-29. Operation Example When Interrupt Is Generated After A/D Conversion Ends**

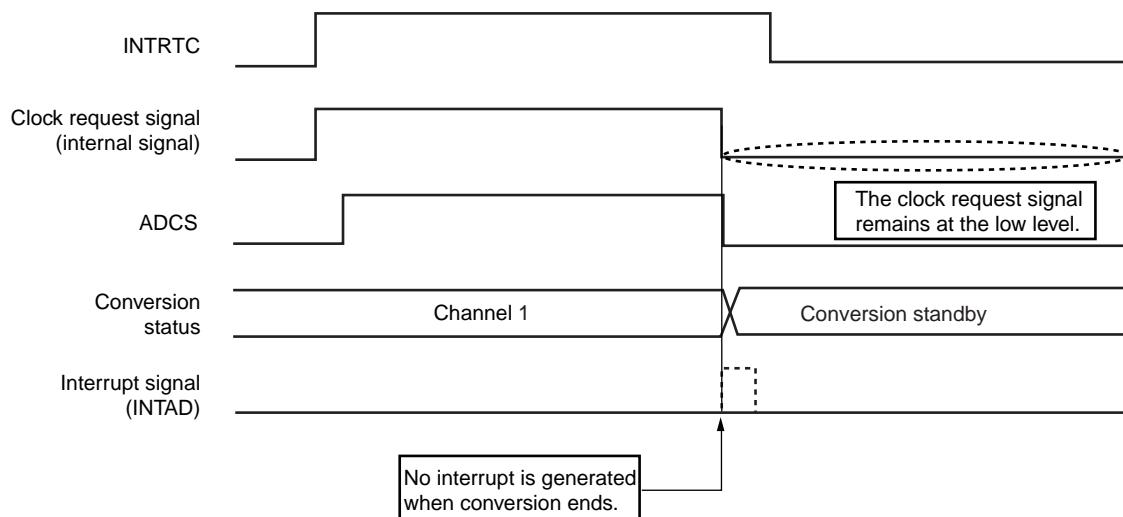


**(2) If no interrupt is generated after A/D conversion ends**

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.

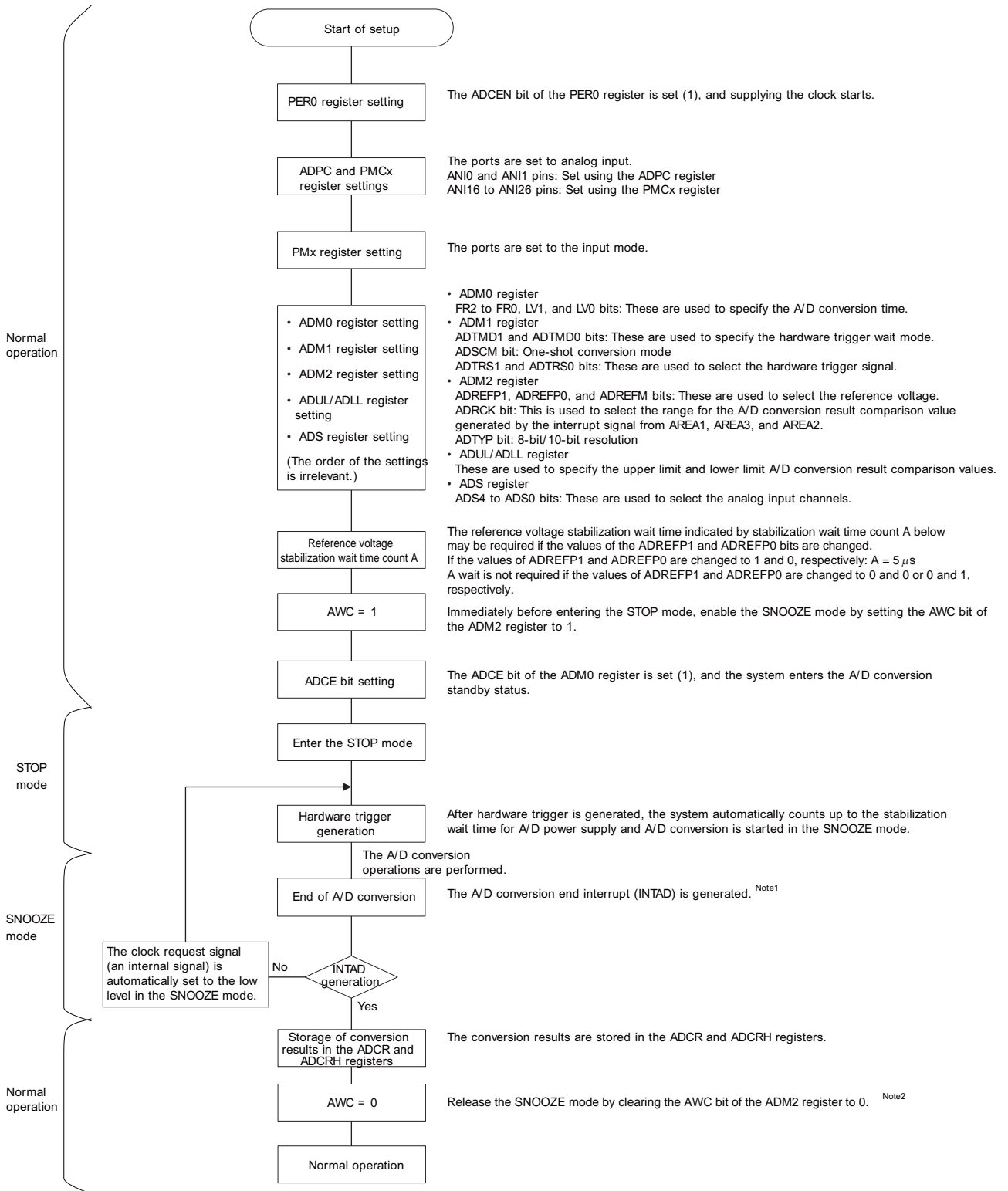
If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

**Figure 12-30. Operation Example When No Interrupt Is Generated After A/D Conversion Ends**



<R>

Figure 12-31. Flowchart for Setting up SNOOZE Mode



- Notes**
- If the A/D conversion end interrupt request signal (INTAD) is not generated by setting ADRCCK 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.
  - 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.

### 12.9 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.

$$1\text{LSB} = 1/2^{10} = 1/1024$$

$$= 0.098\% \text{FSR}$$

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.

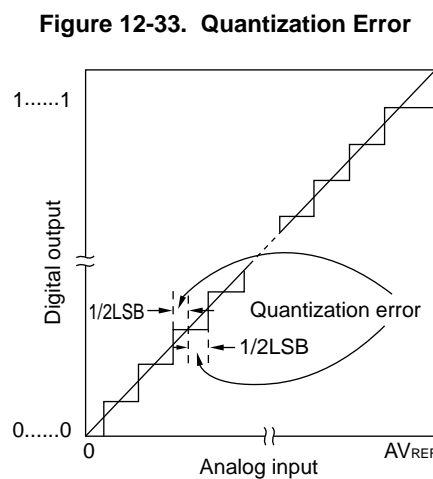
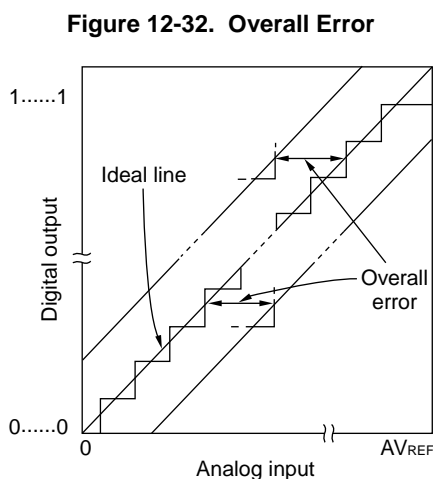
Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

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  $\pm 1/2\text{LSB}$  error naturally occurs. In an A/D converter, an analog input voltage in a range of  $\pm 1/2\text{LSB}$  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.



#### (4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value ( $1/2\text{LSB}$ ) 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 ( $3/2\text{LSB}$ ) 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.

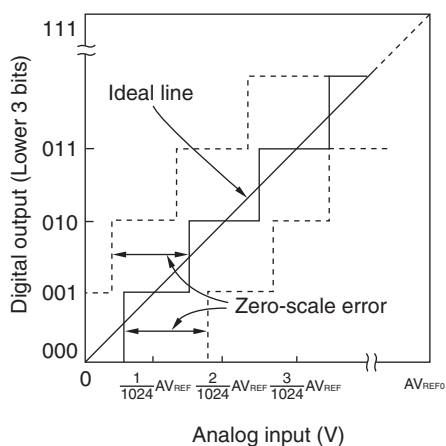
**(6) Integral linearity error**

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.

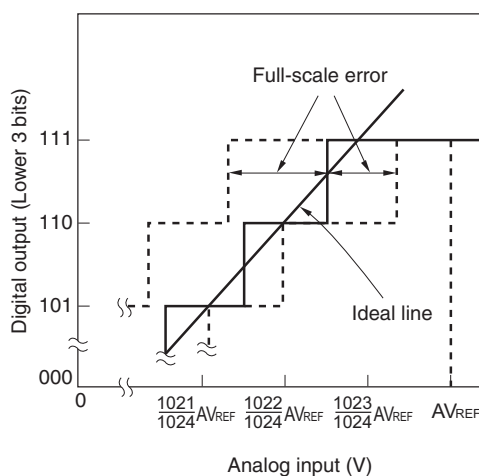
**(7) Differential linearity error**

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

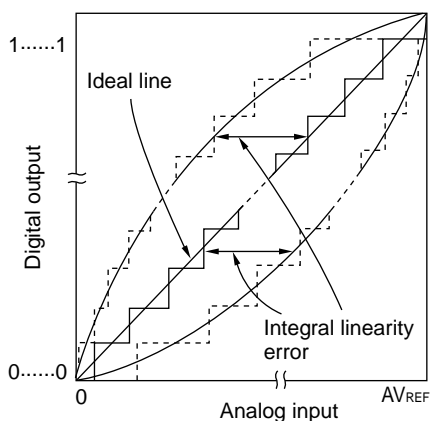
**Figure 12-34. Zero-Scale Error**



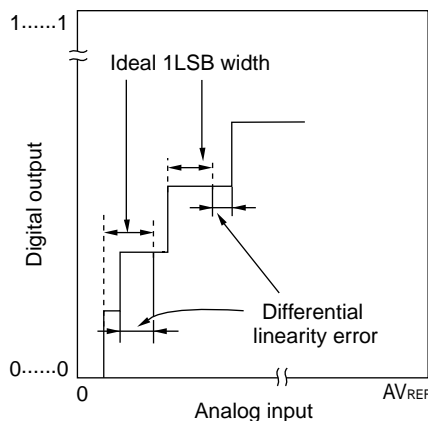
**Figure 12-35. Full-Scale Error**



**Figure 12-36. Integral Linearity Error**



**Figure 12-37. Differential Linearity Error**

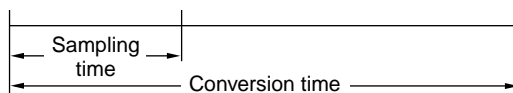


**(8) Conversion time**

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.



## 12.10 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, ANI1 and ANI16 to ANI25 pins

Observe the rated range of the ANI0, ANI1 and ANI16 to ANI25 pins input voltage. If a voltage exceeding  $V_{DD}$  and  $AV_{REFP}$  or below  $V_{SS}$  and  $AV_{REFM}$  (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 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** Internal reference voltage (1.45 V) can be used 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, ANI1, and ANI16 to ANI25 pins.

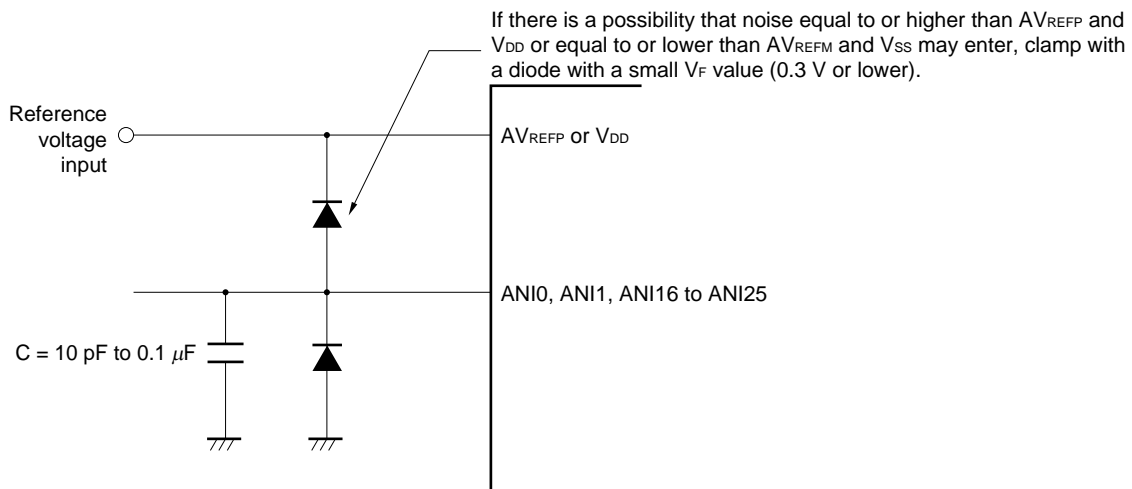
<1> Connect a capacitor with a low equivalent resistance and a good frequency response (capacitance of about 0.01  $\mu$ F) via the shortest possible run of relatively thick wiring to the power supply.

<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 12-38 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 12-38. Analog Input Pin Connection



### (5) Analog input (ANIn) pins

- <1> The analog input pins (ANI0, ANI1) are also used as input port pins (P20, P21). When A/D conversion is performed with any of the ANI0, ANI1 pins selected, do not change to output value P20, P21 while conversion is in progress; otherwise the conversion resolution may be degraded.
- <2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to avoid the input or output of digital signals and signals with similarly sharp transitions during 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, we recommend using the converter with analog input sources that have output impedances no greater than 1 k $\Omega$ . If the output impedance cannot be set to 1 k $\Omega$  or smaller, lengthen the sampling time or connect a larger capacitor (with a value of about 0.1  $\mu$ F) to the pin from among ANI0, ANI1, and ANI16 to ANI25 to which the source is connected (see **Figure 12-38**). 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.

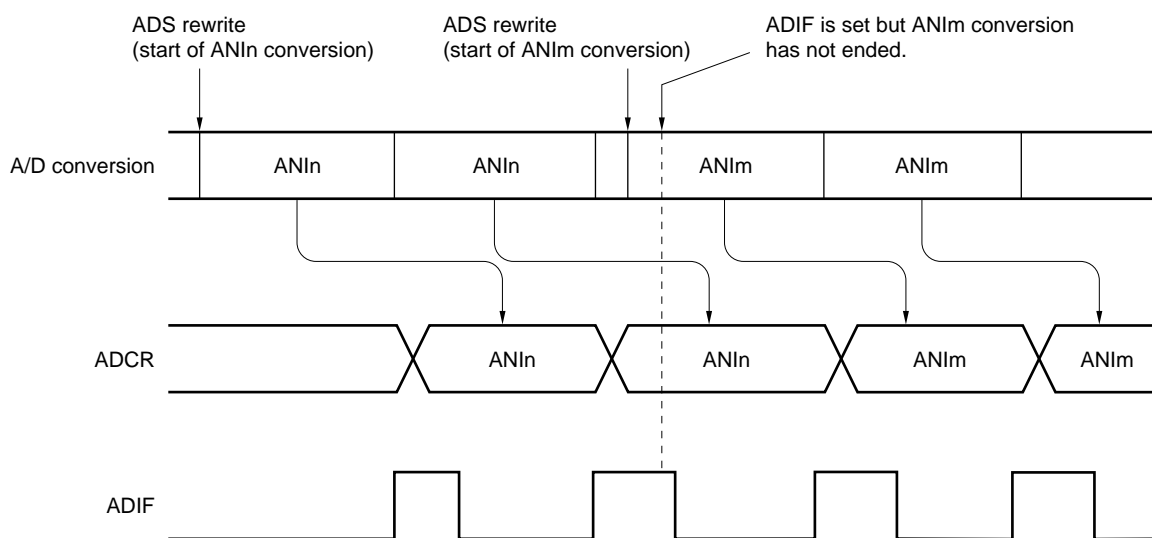
**(7) Interrupt request flag (ADIF)**

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 12-39. 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  $\mu$ s 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.

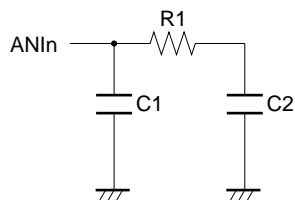
**(9) A/D conversion result register (ADCR, ADCRH) read operation**

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 12-40. Internal Equivalent Circuit of ANIn Pin**



**Table 12-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)**

$AV_{REFP}, V_{DD}$	ANIn Pins	R1 [k $\Omega$ ]	C1 [pF]	C2 [pF]
$3.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	ANI0, ANI1	14	8	2.5
	ANI16 to ANI25	18	8	7.0
$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	ANI0, ANI1	39	8	2.5
	ANI16 to ANI25	53	8	7.0
$1.8\text{ V} \leq V_{DD} \leq 2.7\text{ V}$	ANI0, ANI1	231	8	2.5
	ANI16 to ANI25	321	8	7.0
$1.6\text{ V} \leq V_{DD} < 2.7\text{ V}$	ANI0, ANI1	632	8	2.5
	ANI16 to ANI25	902	8	7.0

**Remark** The resistance and capacitance values shown in Table 12-4 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 13 COMPARATOR

The comparator compares a reference input voltage to an analog input voltage. It consists of two independent comparators: comparator 0 and comparator 1.

### 13.1 Functions of Comparator

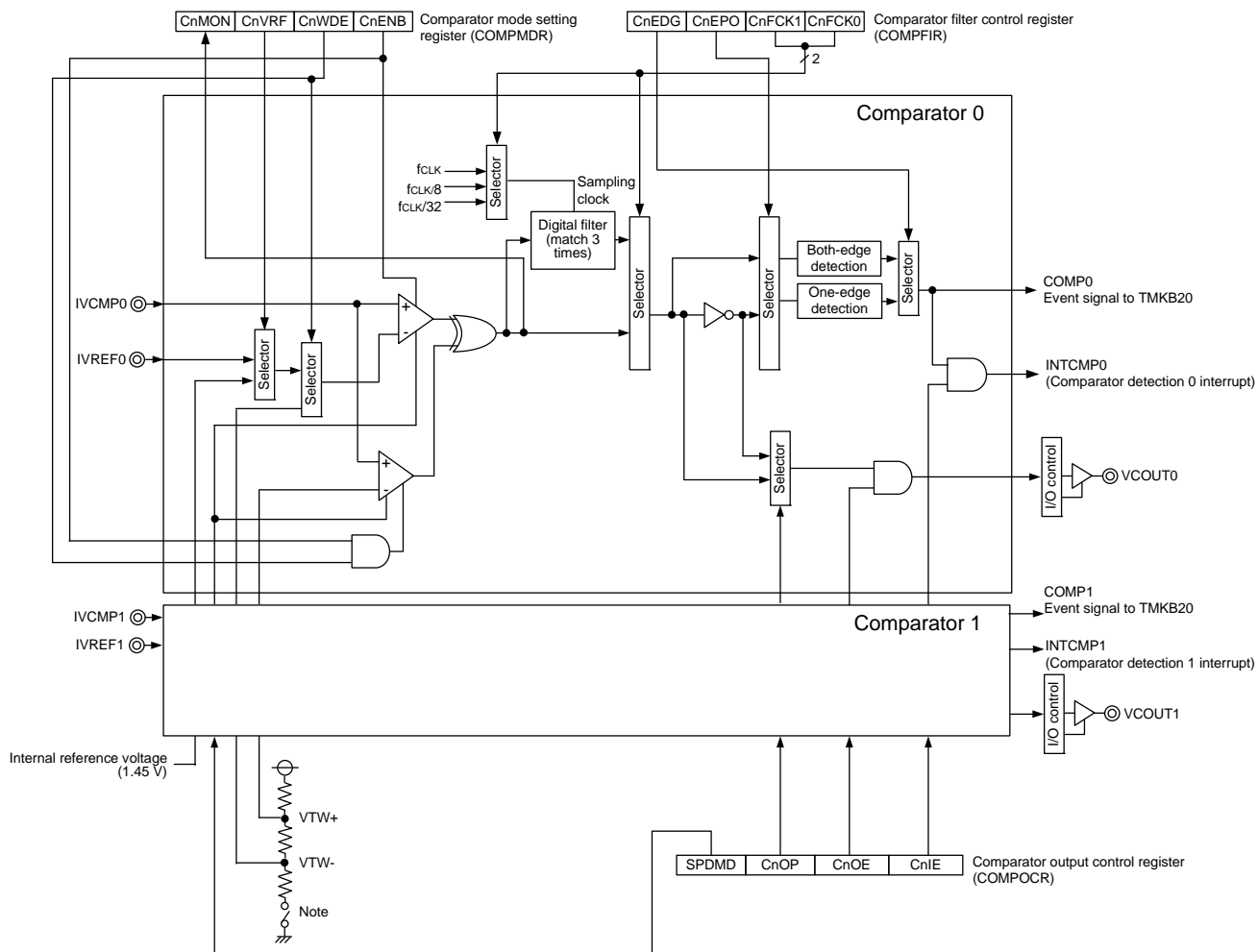
The comparator has the following functions.

- Comparator high-speed mode, comparator low-speed mode, or comparator window mode can be selected.
- The external reference voltage input or internal reference voltage can be selected as the reference voltage.
- The canceling width of the noise canceling digital filter can be selected.
- An interrupt signal can be generated by detecting an active edge of the comparator output.
- An event signal can be output to TMKB20 by detecting an active edge of the comparator output.

### 13.2 Configuration of Comparator

Figure 13-1 shows the comparator block diagram.

**Figure 13-1. Comparator Block Diagram**



**Note** When either or both of the C0WDE and C1WDE bits are set to 1, this switch is turned on and the divider resistors for generating the comparison voltage are enabled.

**Remark** n = 0, 1

### 13.3 Registers Controlling Comparator

Table 13-1 lists the registers controlling comparator.

**Table 13-1. Registers Controlling Comparator**

Register Name	Symbol
Peripheral enable register 1	PER1
Comparator mode setting register	COMPMDR
Comparator filter control register	COMPFIR
Comparator output control register	COMPOCR
Port mode control register 4	PMC4
LCD port function register 3	PFSEG3
Port mode registers 0, 4	PM0, PM4
Port registers 0, 4	P0, P4

#### 13.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the Comparator is used, be sure to set bit 5 (CMPEN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 13-2. Format of Peripheral Enable Register 1 (PER1)**

Address: F007AH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
PER1	TMKAEN	0	CMPEN	TKB2EN	0	0	0	0

CMPEN	Control of comparator input clock
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by the Comparator cannot be written.</li> <li>• The Comparator is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the Comparator can be read/written.</li> </ul>

- Cautions**
1. When setting the comparator, be sure to set the CMPEN bit to 1 first and the following registers. If CMPEN = 0, a control register of the comparator becomes default value, and all write values are ignored (except for port mode registers 0, 4 (PM0, PM4), port registers 0, 4 (P0, P4), port mode control register 4 (PMC4), and LCD port function register 3 (PFSEG3)).
    - Comparator mode setting register (COMPMDR)
    - Comparator filter control register (COMPFIR)
    - Comparator output control register (COMPOCR)
  2. Be sure to clear the bits 6 and 3 to 0 to "0".

13.3.2 Comparator mode setting register (COMPMDR)

Figure 13-3. Format of Comparator Mode Setting Register (COMPMDR)

Address: F0340H After reset: 00H R/W

Symbol	<7>	6	5	<4>	<3>	2	1	<0>
COMPMDR	C1MON	C1VRF	C1WDE	C1ENB	C0MON	C0VRF	C0WDE	C0ENB

C1MON	Comparator 1 monitor flag <sup>Notes 3, 7</sup>
0	In standard mode: IVCMP1 < comparator 1 reference voltage or comparator 1 stopped In window mode: IVCMP1 < low-voltage reference or IVCMP1 > high-voltage reference
1	In standard mode: IVCMP1 > comparator 1 reference voltage In window mode: Low-voltage reference < IVCMP1 < high-voltage reference

C1VRF	Comparator 1 reference voltage selection <sup>Notes 1, 4, 5, 6</sup>
0	Comparator 1 reference voltage is IVREF1 input
1	Comparator 1 reference voltage is internal reference voltage (1.45 V)

C1WDE	Comparator 1 window mode selection <sup>Note 2</sup>
0	Comparator 1 standard mode
1	Comparator 1 window mode

C1ENB	Comparator 1 operation enable
0	Comparator 1 operation disabled
1	Comparator 1 operation enabled

C0MON	Comparator 0 monitor flag <sup>Notes 3, 7</sup>
0	In standard mode: IVCMP0 < comparator 0 reference voltage or comparator 0 stopped In window mode: IVCMP0 < low-voltage reference or IVCMP0 > high-voltage reference
1	In standard mode: IVCMP0 > comparator 0 reference voltage In window mode: Low-voltage reference < IVCMP0 < high-voltage reference

(Notes are listed on the next page.)

C0VRF	Comparator 0 reference voltage selection <sup>Notes 1, 4, 5, 6</sup>
0	Comparator 0 reference voltage is IVREF0 input
1	Comparator 0 reference voltage is internal reference voltage (1.45 V)

C0WDE	Comparator 0 window mode selection <sup>Note 2</sup>
0	Comparator 0 standard mode
1	Comparator 0 window mode

C0ENB	Comparator 0 operation enable
0	Comparator 0 operation disabled
1	Comparator 0 operation enabled

- Notes**
- Valid only when standard mode is selected. In window mode, the reference voltage in the comparator is selected regardless of the setting of this bit.
  - Window mode cannot be set when low-speed mode is selected (the SPDMD bit in the COMPOCR register is 0).
  - The initial value is 0 immediately after a reset is released. However, the value is undefined when C0ENB is set to 0 and C1ENB is set to 0 after operation of the comparator is enabled once.
  - The internal reference voltage (1.45 V) can be selected in HS (high-speed main) mode. When the internal reference voltage (1.45 V) is selected in HS (high-speed main) mode, the temperature sensor output cannot be A/D converted.
  - Do not select the internal reference voltage in STOP mode.
  - Do not select the internal reference voltage when the subsystem clock ( $f_{XT}$ ) is selected as the CPU clock and both the high-speed system clock ( $f_{MX}$ ) and high-speed on-chip oscillator clock ( $f_{IH}$ ) are stopped.
  - Writing to this bit is ignored.

## 13.3.3 Comparator filter control register (COMPFIR)

Figure 13-4. Format of Comparator Filter Control Register (COMPFIR)

Address: F0341H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0	
COMPFIR	C1EDG	C1EPO	C1FCK1	C1FCK0	C0EDG	C0EPO	C0FCK1	C0FCK0	
C1EDG	Comparator 1 edge detection selection <sup>Note 1</sup>								
0	Interrupt request by comparator 1 one-edge detection								
1	Interrupt request by comparator 1 both-edge detection								
C1EPO	Comparator 1 edge polarity switching <sup>Note 1</sup>								
0	Interrupt request at comparator 1 rising edge								
1	Interrupt request at comparator 1 falling edge								
C1FCK1	C1FCK0	Comparator 1 filter selection <sup>Note 1</sup>							
0	0	No comparator 1 filter							
0	1	Comparator 1 filter enabled, sampling at $f_{CLK}$							
1	0	Comparator 1 filter enabled, sampling at $f_{CLK}/8$							
1	1	Comparator 1 filter enabled, sampling at $f_{CLK}/32$							
C0EDG	Comparator 0 edge detection selection <sup>Note 2</sup>								
0	Interrupt request by comparator 0 one-edge detection								
1	Interrupt request by comparator 0 both-edge detection								
C0EPO	Comparator 0 edge polarity switching <sup>Note 2</sup>								
0	Interrupt request at comparator 0 rising edge								
1	Interrupt request at comparator 0 falling edge								
C0FCK1	C0FCK0	Comparator 0 filter selection <sup>Note 2</sup>							
0	0	No comparator 0 filter							
0	1	Comparator 0 filter enabled, sampling at $f_{CLK}$							
1	0	Comparator 0 filter enabled, sampling at $f_{CLK}/8$							
1	1	Comparator 0 filter enabled, sampling at $f_{CLK}/32$							

- Notes 1.** If bits C1FCK1, C1FCK0, C1EPO, and C1EDG are changed, a comparator 1 interrupt request and an event signal to TMKB20 may be generated. Change these bits only after setting 16-bit timer KB2 counter restart select register 09 (ELSELR09) to 0, and bits 5 and 13 of forced output stop function control registers 0, 1 (TKBPACTL00, TKBPACTL01) to 0. Also, be sure to clear (0) bit 7 (CMPIF1) in interrupt request flag register 2L (IF2L). If bits C1FCK1 and C1FCK0 are changed from 00B (no comparator 1 filter) to a value other than 00B (comparator 1 filter enabled), allow four sampling times to elapse until the filter output is updated, and then use the comparator 1 interrupt request or the event signal to TMKB20.
- 2.** If bits C0FCK1, C0FCK0, C0EPO, and C0EDG are changed, a comparator 0 interrupt request and an event signal to TMKB20 may be generated. Change these bits only after setting 16-bit timer KB2 counter restart select register 08 (ELSELR08) to 0, and bits 4 and 12 of forced output stop function control registers 0, 1 (TKBPACTL00, TKBPACTL01) to 0. Also, be sure to clear (0) bit 6 (CMPIF0) in interrupt request flag register 2L (IF2L). If bits C0FCK1 and C0FCK0 are changed from 00B (no comparator 0 filter) to a value other than 00B (comparator 0 filter enabled), allow four sampling times to elapse until the filter output is updated, and then use the comparator 0 interrupt request or the event signal to TMKB20.

## 13.3.4 Comparator output control register (COMPOCR)

Figure 13-5. Format of Comparator Output Control Register (COMPOCR)

Address: F0342H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
COMPOCR	SPDMD	C1OP	C1OE	C1IE	0	C0OP	C0OE	C0IE

SPDMD	Comparator speed selection <sup>Note 1</sup>
0	Comparator low-speed mode
1	Comparator high-speed mode

C1OP	VCOU1 output polarity selection
0	Comparator 1 output is output to VCOU1
1	Inverted comparator 1 output is output to VCOU1

C1OE	VCOU1 pin output enable
0	Comparator 1 VCOU1 pin output disabled
1	Comparator 1 VCOU1 pin output enabled

C1IE	Comparator 1 interrupt request enable <sup>Note 2</sup>
0	Comparator 1 interrupt request disabled
1	Comparator 1 interrupt request enabled

C0OP	VCOU0 output polarity selection
0	Comparator 0 output is output to VCOU0
1	Inverted comparator 0 output is output to VCOU0

C0OE	VCOU0 pin output enable
0	Comparator 0 VCOU0 pin output disabled
1	Comparator 0 VCOU0 pin output enabled

C0IE	Comparator 0 interrupt request enable <sup>Note 3</sup>
0	Comparator 0 interrupt request disabled
1	Comparator 0 interrupt request enabled

- Notes**
- When rewriting the SPDMD bit, be sure to set the CiENB bit (i = 0 or 1) in the COMPMDR register to 0 in advance.
  - If C1IE is changed from 0 (interrupt requests disabled) to 1 (interrupt requests enabled), bit 7 (CMPIF1) in interrupt request flag register 2L (IF2L) might be set to 1 (interrupt requested), so be sure to clear (0) bit 7 (CMPIF1) in interrupt request flag register 2L (IF2L) before using interrupts.
  - If C0IE is changed from 0 (interrupt requests disabled) to 1 (interrupt requests enabled), bit 6 (CMPIF0) in interrupt request flag register 2L (IF2L) might be set to 1 (interrupt requested), so be sure to clear (0) bit 6 (CMPIF0) in interrupt request flag register 2L (IF2L) before using interrupts.

### 13.3.5 Registers controlling port functions of analog input pins

When using the IVREF0 pin for analog input of the comparator, set bit 2 of LCD port function register 3 (PFSEG3), bit 5 of the port mode register 4 (PM4) and the port mode control register 4 (PMC4) to 1.

When using the IVCMP0, IVCMP1, and IVREF1 pins for analog input of the comparator, set the port mode register (PMxx) and the port mode control register (PMCxx) bit corresponding to each port to 1.

When using the VCOU0 and VCOU1 functions, set the registers (port mode register (PMxx) and port register (Pxx) that control the port functions shared with the target channels.

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, and **4.3.9 LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)**.



### 13.4 Operation

Comparator 0 and comparator 1 operate independently. Their setting methods are not the same but operations are the same. Table 13-2 lists the Procedure for Setting Comparator Associated Registers.

**Table 13-2. Procedure for Setting Comparator Associated Registers**

Step	Register	Bit	Setting Value
1	PER1	CMPEN	1 (input clock supply)
2	PFSEG3	PFDEG	Select the function of pins IVCMPi and IVREFi.
	PMC4	PMC4n	Set the PFDEG bit to 1 (digital input invalid) (only when IVREF0 is used).
	PM4	PM4n	Set the PMC4n bit to 1 (analog input). Set the PM4n bit to 1 (input mode).
3	COMPOCR	SPDMD	Select the comparator response speed (0: Low-speed mode/1: High-speed mode), <sup>Note 1</sup>
4	COMPMDR	CiWDE	0 (standard mode)   1 (window mode) <sup>Note 2</sup>
		CiVRF	0 (Reference = IVREFi input)   1 (Reference = internal reference voltage (1.45 V)) <sup>Note 4</sup>   Window comparator operation (reference = internal VREF)
		CiENB	1 (operation enabled)
5	Wait for comparator stabilization time $t_{CMP}$		
6	COMPFIR	CiFCK1, CiFCK0	Select whether the digital filter is used or not and the sampling clock.
		CiEPO, CiEDG	Select the edge detection condition for an interrupt request (rising edge/falling edge/both edges).
7	COMPOCR	CiOP, CiOE	Set the VCOUTi output (select the polarity and set output enabled or disabled). See 13.4.3 <b>Comparator i output (i = 0 or 1)</b> .
		CiIE	Set the interrupt request output enabled or disabled. See 13.4.3 <b>Comparator i output (i = 0 or 1)</b> .
8	PR2L	CMPPR0i, CMPPR1i	When using an interrupt: Select the interrupt priority level.
9	MK2L	CMPMKi	When using an interrupt: Select the interrupt masking.
10	IF2L	CMPIFi	When using an interrupt: 0 (no interrupt requested: initialization) <sup>Note 3</sup>

- Notes**
1. Comparator 0 and comparator 1 cannot be set independently.
  2. Can be set in high-speed mode (SPDMD = 1).
  3. After the setting of the comparator, an unnecessary interrupt may occur until operation becomes stable, so initialize the interrupt flag.
  4. Can be set in HS (high-speed main) mode.

**Remark** i = 0, 1, n = 2 to 5

Figures 13-6 and 13-7 show comparator i (i = 0 or 1) operation examples. In standard mode, the CiMON bit in the COMPMDR register is set to 1 when the analog input voltage is higher than the reference input voltage, and the CiMON bit is set to 0 when the analog input voltage is lower than the reference input voltage.

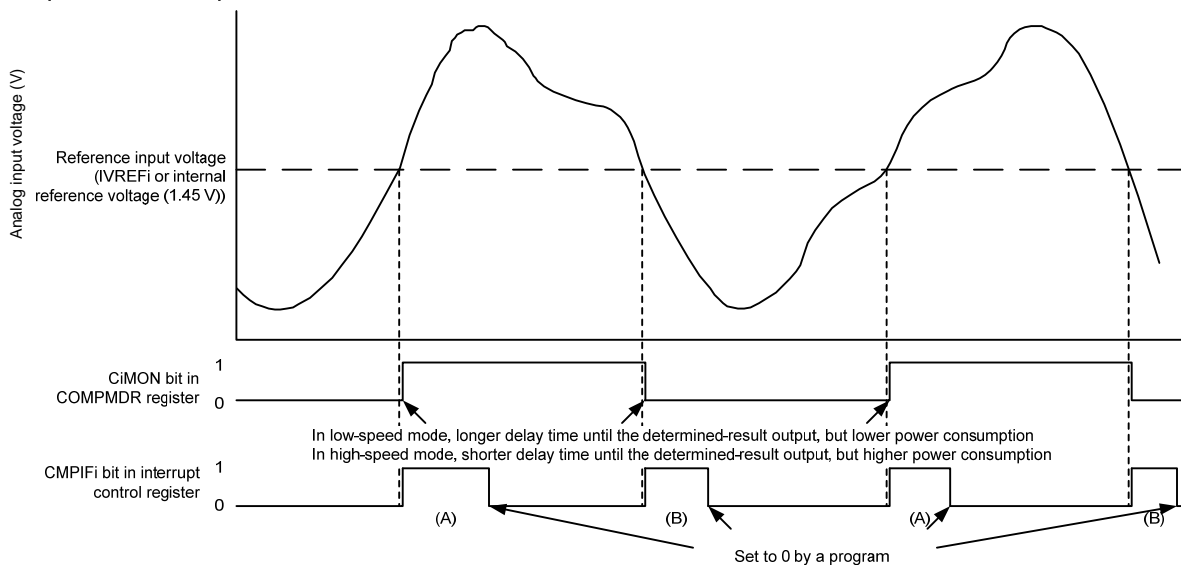
In window mode, the CiMON bit in the COMPMDR register is set to 1 when the analog input voltage meets the following condition, and the CiMON bit is set to 0 when the analog input voltage does not meet the following condition:

“Low-voltage reference voltage < analog input voltage < high-voltage reference voltage”

When using the comparator i interrupt, set CiIE in the COMPOCR register to 1 (interrupt request enabled). If the comparison result changes at this time, a comparator i interrupt request is generated. For details on interrupt requests, see 13.4.2 **Comparator i (i = 0 or 1) interrupts**.

Figure 13-6. Example of Comparator i (i = 0 or 1) Operation in Standard Mode

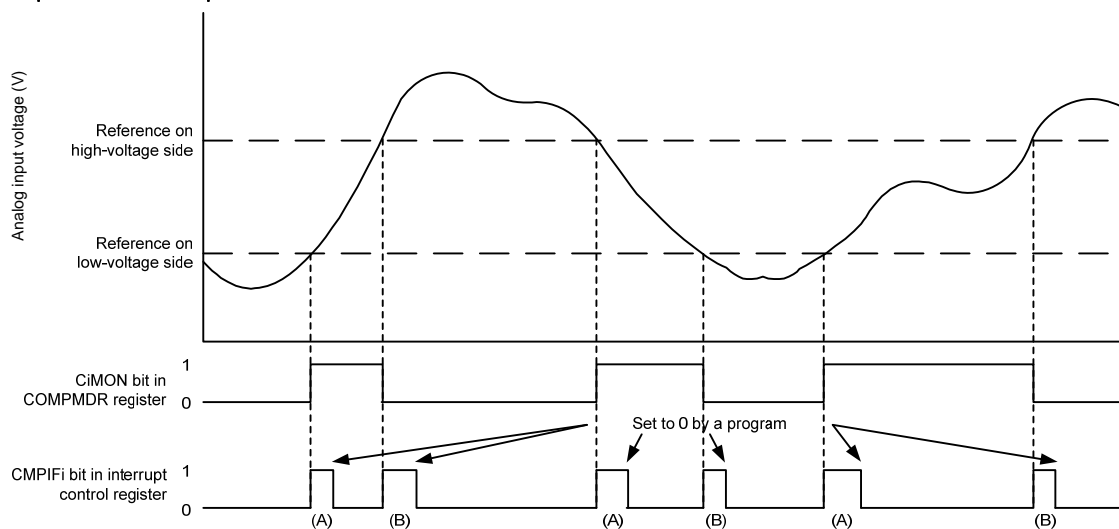
• Operation example in standard mode



**Caution** The above diagram applies when CiFCK1 and CiFCK0 in the COMPFIR register = 00B (no filter) and CiEDG = 1 (both edges). When CiEDG = 0 and CiEPO = 0 (rising edge), CMPIFi changes as shown by (A) only. When CiEDG = 0 and CiEPO = 1 (falling edge), CMPIFi changes as shown by (B) only.

Figure 13-7. Example of Comparator i (i = 0 or 1) Operation in Window Mode

• Operation example in window mode



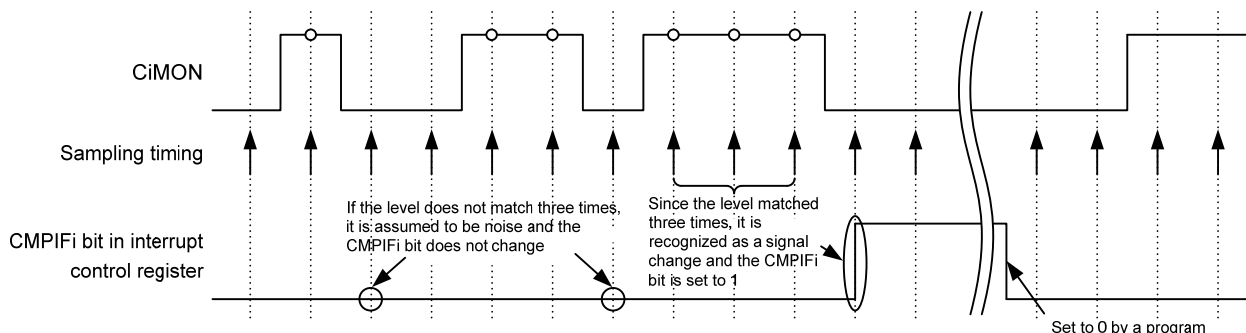
**Caution** The above diagram applies when CiFCK1 and CiFCK0 in the COMPFIR register = 00B (no filter) and CiEDG = 1 (both edges). When CiEDG = 0 and CiEPO = 0 (rising edge), CMPIFi changes as shown by (A) only. When CiEDG = 0 and CiEPO = 1 (falling edge), CMPIFi changes as shown by (B) only.

### 13.4.1 Comparator i digital filter (i = 0 or 1)

Comparator i contains a digital filter. The sampling clock can be selected by bits CiFCK1 and CiFCK0 in the COMPFIR register. The comparator i output signal is sampled every sampling clock, and when the level matches three times, that value is determined as the digital filter output at the next sampling clock.

Figure 13-8 shows the comparator i (i = 0 or 1) digital filter and interrupt operation example.

Figure 13-8. Comparator i (i = 0 or 1) Digital Filter and Interrupt Operation Example



**Caution** The above operation example applies when bits CiFCK1 and CiFCK0 in the COMPFIR register is 01B, 10B, or 11B (digital filter enabled).

### 13.4.2 Comparator i (i = 0 or 1) interrupts

The comparator generates interrupt requests from two sources, comparator 0 and comparator 1. The comparator i interrupt each uses a priority level specification flag, an interrupt mask flag, an interrupt request flag, and a single vector.

When using the comparator i interrupt, set the CiIE bit in the COMPOCR register to 1 (interrupt request output enabled). The condition for interrupt request generation can be set by the COMPFIR register. The comparator outputs can also be passed through the digital filter. Three different sampling clocks can be selected for the digital filter.

For details on the register setting and interrupt request generation, see 13.3.3 Comparator filter control register (COMPFIR) and 13.3.4 Comparator output control register (COMPOCR).

### 13.4.3 Comparator i output (i = 0 or 1)

The comparison result from the comparator can be output to external pins. Bits CiOP and CiOE in the COMPOCR register can be used to set the output polarity (non-inverted output or inverted output) and output enabled or disabled. For the correspondence between the register setting and the comparator output, see **13.3.4 Comparator output control register (COMPOCR)**.

To output the comparator comparison result to the VCOUTi output pin, use the following procedure to set the ports. Note that the ports are set to input after reset.

- <1> Set the mode for the comparator (Steps 2 to 5 as listed in **Table 13-2 Procedure for Setting Comparator Associated Registers**).
- <2> Set the VCOUTi output for the comparator (set the COMPOCR register to select the polarity and enable the output).
- <3> Set the corresponding port mode control register bit for the VCOUTi output pin to 0.
- <4> Set the corresponding port register bit for the VCOUTi output pin to 0.
- <5> Set the corresponding port direction register for the VCOUTi output pin to output (start outputting from the pin).

### 13.4.4 Stopping or supplying comparator clock

To stop the comparator clock by setting peripheral enable register 1 (PER1), use the following procedure:

- <1> Set the CiENB bit in the COMPMDR register to 0 (stop the comparator).
- <2> Set the CMPIFi bit in registers IF2L to 0 (clear any unnecessary interrupt before stopping the comparator).
- <3> Set the CMPEN bit in the PER1 register to 0.

When the clock is stopped by setting PER1, all the internal registers in the comparator are initialized. To use the comparator again, follow the procedure in Table 13-2 to set the registers.

**Caution** The temperature sensor output cannot be A/D converted while the comparator n reference voltage select bit (CnVRF) in the comparator mode setting register (COMPMDR) is 1 (comparator n reference voltage is internal reference voltage (1.45 V)). (n = 0, 1)

## CHAPTER 14 SERIAL ARRAY UNIT

Serial array unit has up to four serial channels. Each channel can achieve 3-wire serial (CSI), UART, and simplified I<sup>2</sup>C communication.

Function assignment of each channel supported by the RL78/L13 is as shown below.

- 64-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I <sup>2</sup> C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	UART2	–
	1	–		–

- 80-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I <sup>2</sup> C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	UART2	–
	1	–		–
	2	–	UART3	–
	3	–		–

When “UART0” is used for channels 0 and 1 of the unit 0, CSI00 and IIC00 cannot be used, but CSI10, UART1, or IIC10 can be used.

**Caution** Most of the following descriptions in this chapter use the units and channels of the 80-pin products as an example.

## 14.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/L13 has the following features.

### 14.1.1 3-wire serial I/O (CSI00, CSI10)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel.

3-wire serial 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 **14.5 Operation of 3-Wire Serial I/O (CSI00, CSI10) 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<sup>Note</sup>

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 32 ELECTRICAL SPECIFICATIONS ( $T_A = -40$  to  $+85^\circ\text{C}$ )** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS ( $T_A = -40$  to  $+105^\circ\text{C}$ )**.

### 14.1.2 UART (UART0 to UART3)

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). The LIN-bus can be implemented by using timer array unit with an external interrupt (INTP0).

For details about the settings, see **14.6 Operation of UART (UART0 to UART3) Communication**.

[Data transmission/reception]

- Data length of 7, 8, or 9 bits<sup>Note</sup>
- Select the MSB/LSB first
- 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

In addition, UART0 and UART2 reception support the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only UART0, UART2 can be specified for asynchronous reception.

The LIN-bus is accepted in UART0 (0 and 1 channels of unit 0).

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit

**Note** Only UART0, UART2 can be specified for the 9-bit data length.

### 14.1.3 Simplified I<sup>2</sup>C (IIC00, IIC10)

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 I<sup>2</sup>C 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 **14.8 Operation of Simplified I<sup>2</sup>C (IIC00, IIC10) Communication.**

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function<sup>Note</sup> 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]

- ACK error, or overrun error

\* [Functions not supported by simplified I<sup>2</sup>C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait 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 **14.8.3 (2)** for details.

**Remarks 1.** To use an I<sup>2</sup>C bus of full function, see **CHAPTER 15 SERIAL INTERFACE IICA.**

2. m: Unit number (m = 0), n: Channel number (n = 0, 2)



## 14.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

**Table 14-1. Configuration of Serial Array Unit**

Item	Configuration
Shift register	8 bits or 9 bits <sup>Note 1</sup>
Buffer register	Lower 8 bits or 9 bits of serial data register mn (SDRmn) <sup>Notes 1, 2</sup>
Serial clock I/O	SCK00, SCK10 pins (for 3-wire serial I/O), SCL00, SCL10 pins (for simplified I <sup>2</sup> C)
Serial data input	SI00, SI10 pins (for 3-wire serial I/O), RxD1 to RxD3 pins (for UART), RxD0 pin (for UART supporting LIN-bus)
Serial data output	SO00, SO10 pins (for 3-wire serial I/O), TxD1 to TxD3 pins (for UART), TxD0 pin (for UART supporting LIN-bus)
Serial data I/O	SDA00, SDA10 pins (for simplified I <sup>2</sup> C)
Control registers	<Registers of unit setting block> <ul style="list-style-type: none"> <li>• Peripheral enable register 0 (PER0)</li> <li>• Serial clock select register m (SPSm)</li> <li>• Serial channel enable status register m (SEm)</li> <li>• Serial channel start register m (SSm)</li> <li>• Serial channel stop register m (STm)</li> <li>• Serial output enable register m (SOEm)</li> <li>• Serial output register m (SOM)</li> <li>• Serial output level register m (SOLm)</li> <li>• Serial standby control register m (SSCm)</li> <li>• Input switch control register (ISC)</li> <li>• Noise filter enable register 0 (NFEN0)</li> </ul>
	<Registers of each channel> <ul style="list-style-type: none"> <li>• Serial data register mn (SDRmn)</li> <li>• Serial mode register mn (SMRmn)</li> <li>• Serial communication operation setting register mn (SCRmn)</li> <li>• Serial status register mn (SSRmn)</li> <li>• Serial flag clear trigger register mn (SIRmn)</li> </ul>
	<ul style="list-style-type: none"> <li>• Port input mode registers 0, 1, 3 (PIM0, PIM1, PIM3)</li> <li>• Port output mode registers 0, 1, 3 (POM0, POM1, POM3)</li> <li>• Port mode registers 0, 1, 3 (PM0, PM1, PM3)</li> <li>• Port registers 0, 1, 3 (P0, P1, P3)</li> </ul>

**Notes** 1. The number of bits used as the shift register and buffer register differs depending on the unit and channel.

- mn = 00, 01, 10, 11: lower 9 bits
- Other than above: lower 8 bits

2. 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, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10), q: UART number (q = 0 to 3), r: IIC number (r = 00, 10)

Figure 14-1 shows the block diagram of serial array unit 0.

Figure 14-1. Block Diagram of Serial Array Unit 0

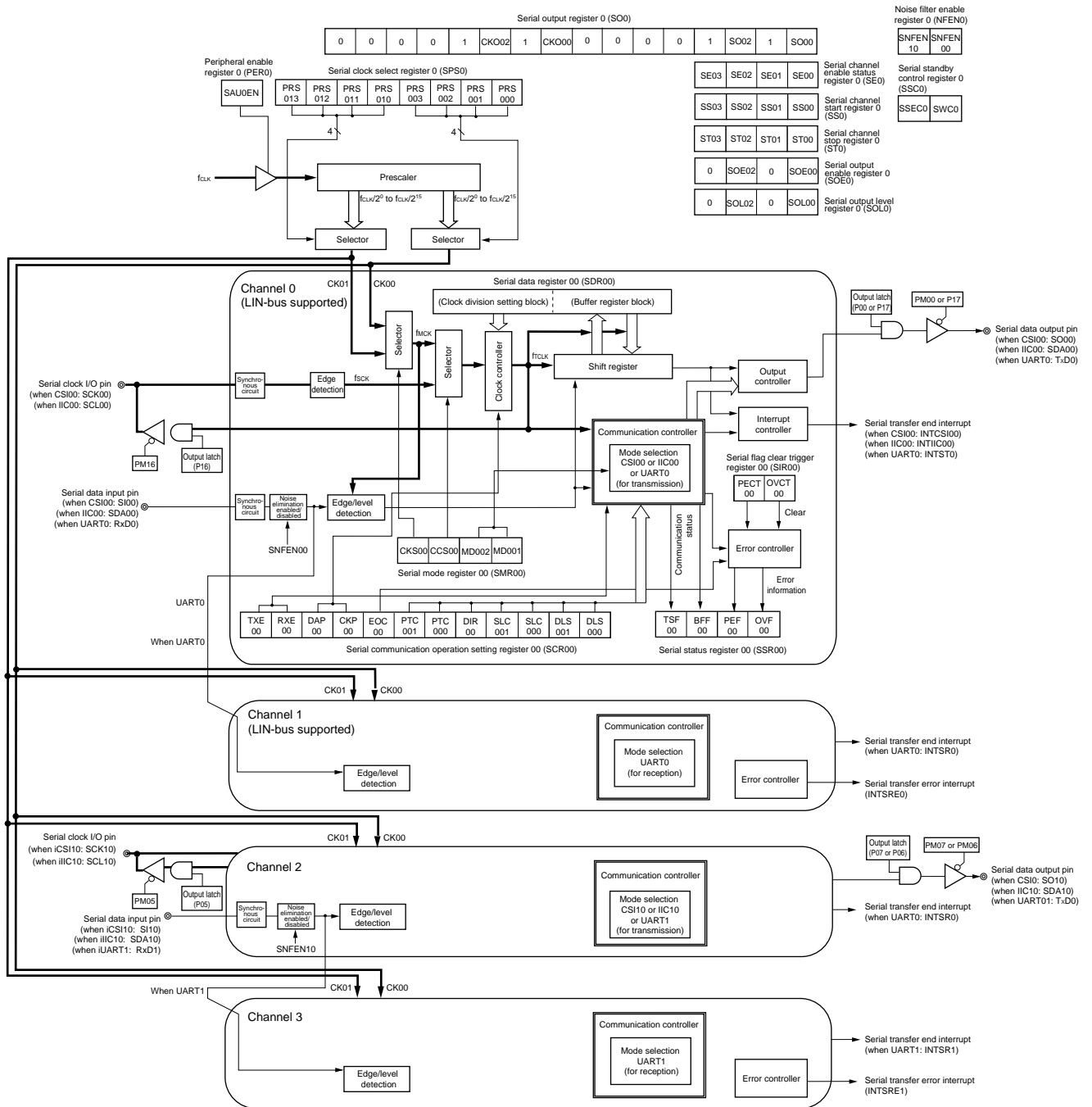
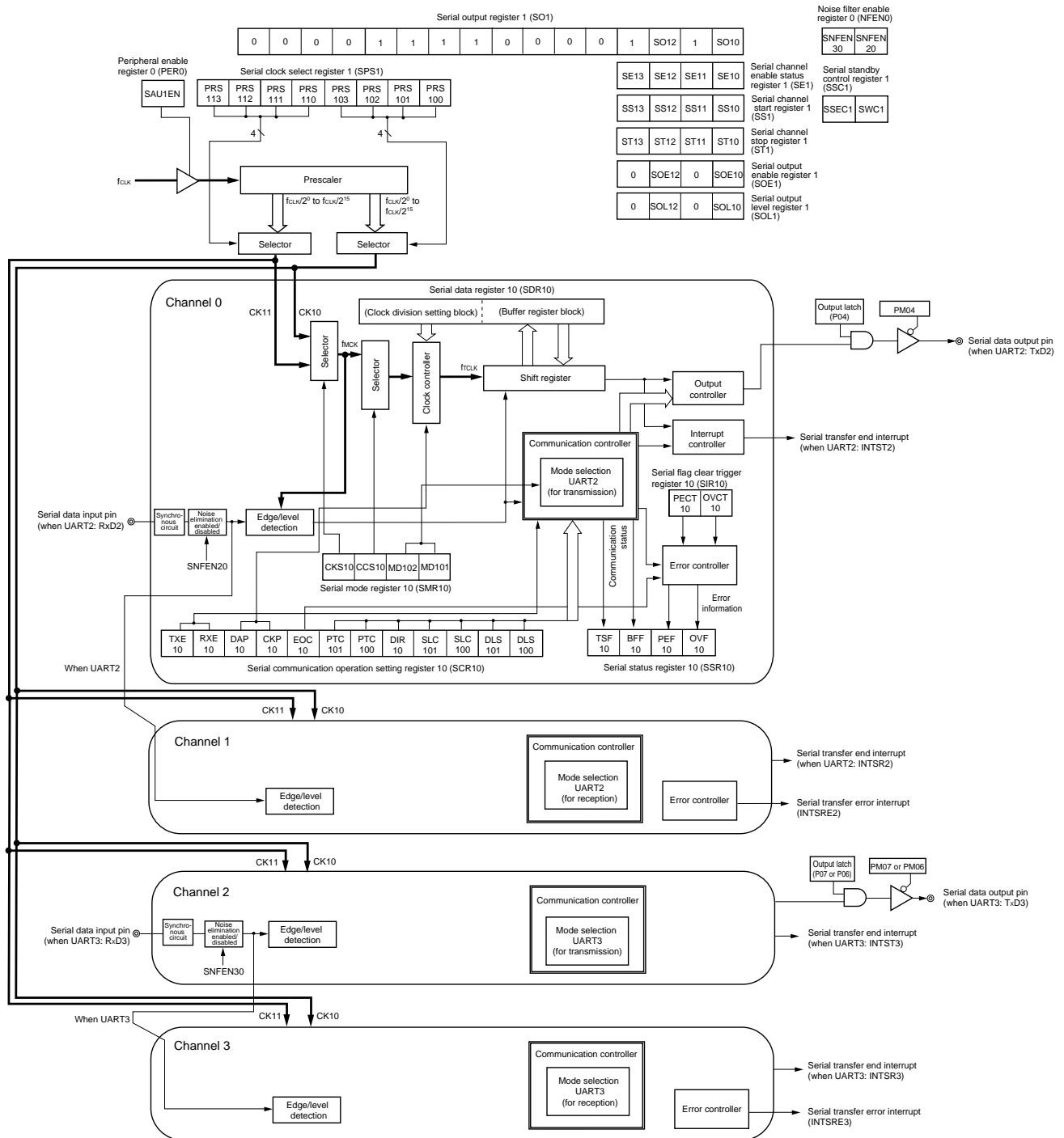


Figure 14-2 shows the block diagram of serial array unit 1.

Figure 14-2. Block Diagram of Serial Array Unit 1 (80-pin products)



### 14.2.1 Shift register

This is a 9-bit register that converts parallel data into serial data or vice versa.

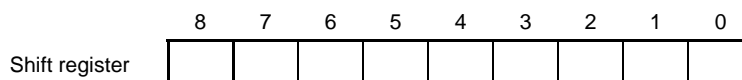
In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used<sup>Note 1</sup>.

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 8/9 bits of serial data register mn (SDRmn).



### 14.2.2 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)<sup>Note 1</sup> 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 8/9 bits. When data is to be transmitted, set transmit to be transferred to the shift register to the lower 8/9 bits.

The data stored in the lower 8/9 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)<sup>Note 1</sup>

The SDRmn register can be read or written in 16-bit units.

The lower 8/9 bits of the SDRmn register can be read or written<sup>Note 2</sup> 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 UART0, UART2 can be specified for the 9-bit data length.

<R> **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.

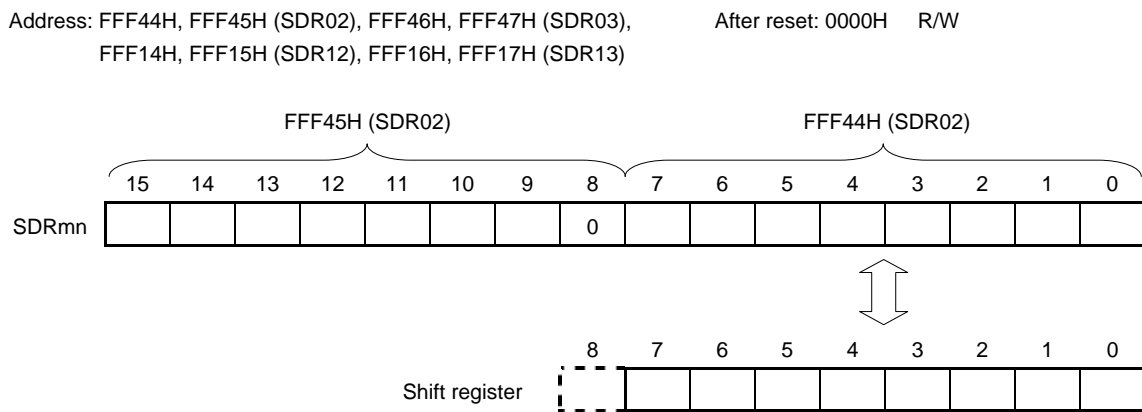
- 2.** m: Unit number ( $m = 0, 1$ ), n: Channel number ( $n = 0$  to 3), p: CSI number ( $p = 00, 10$ ), q: UART number ( $q = 0$  to 3), r: IIC number ( $r = 00, 10$ )

**Figure 14-3. Format of Serial Data Register mn (SDRmn) (mn = 00, 01, 10, 11)**



**Remark** For the function of the higher 7 bits of the SDRmn register, see **14.3 Registers Controlling Serial Array Unit**.

**Figure 14-4. Format of Serial Data Register mn (SDRmn) (mn = 02, 03, 12, 13)**



**Caution** Be sure to clear bit 8 to “0”.

**Remark** For the function of the higher 7 bits of the SDRmn register, see **14.3 Registers Controlling Serial Array Unit**.

### 14.3 Registers Controlling Serial Array Unit

The 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)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 0, 1, 3 (PIM0, PIM1, PIM3)
- Port output mode registers 0, 1, 3 (POM0, POM1, POM3)
- Port mode registers 0, 1, 3 (PM0, PM1, PM3)
- Port registers 0, 1, 3 (P0, P1, P3)

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

### 14.3.1 Peripheral enable register 0 (PER0)

PER0 is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.

When serial array unit 1 is used, be sure to set bit 3 (SAU1EN) 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 14-5. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

SAUmEN	Control of serial array unit m input clock supply
0	Stops supply of input clock. <ul style="list-style-type: none"> <li>• SFR used by serial array unit m cannot be written.</li> <li>• Serial array unit m is in the reset status.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>• SFR used by serial array unit m can be read/written.</li> </ul>

**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 input switch control register (ISC), noise filter enable register 0 (NFEN0), port input mode registers 0, 1, 3 (PIM0, PIM1, PIM3), port output mode registers 0, 1, 3 (POM0, POM1, POM3), port mode registers 0, 1, 3 (PM0, PM1, PM3), and port registers 0, 1, 3 (P0, P1, P3)).

- 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 and 6 to "0".

**14.3.2 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 14-6. Format of Serial Clock Select Register m (SPSm)**

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1) 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) <sup>Note</sup>				
					f <sub>CLK</sub> = 2 MHz	f <sub>CLK</sub> = 5 MHz	f <sub>CLK</sub> = 10 MHz	f <sub>CLK</sub> = 20 MHz	f <sub>CLK</sub> = 24 MHz
0	0	0	0	f <sub>CLK</sub>	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	0	0	1	f <sub>CLK</sub> /2	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	0	1	0	f <sub>CLK</sub> /2 <sup>2</sup>	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	f <sub>CLK</sub> /2 <sup>5</sup>	62.5 kHz	156 kHz	313 kHz	625 kHz	750 kHz
0	1	1	0	f <sub>CLK</sub> /2 <sup>6</sup>	31.3 kHz	78.1 kHz	156 kHz	313 kHz	375 kHz
0	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	187.5 kHz
1	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
1	0	0	1	f <sub>CLK</sub> /2 <sup>9</sup>	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	46.9 kHz
1	0	1	0	f <sub>CLK</sub> /2 <sup>10</sup>	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.4 kHz
1	0	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
1	1	0	0	f <sub>CLK</sub> /2 <sup>12</sup>	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	0	1	f <sub>CLK</sub> /2 <sup>13</sup>	244 Hz	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	1	1	0	f <sub>CLK</sub> /2 <sup>14</sup>	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz
1	1	1	1	f <sub>CLK</sub> /2 <sup>15</sup>	61 Hz	153 kHz	305 Hz	610 Hz	732 Hz

**Note** When changing the clock selected for f<sub>CLK</sub> (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

**Caution** Be sure to clear bits 15 to 8 to “0”.

- Remarks**
1. f<sub>CLK</sub>: CPU/peripheral hardware clock frequency
  2. m: Unit number (m = 0, 1)
  3. k = 0, 1



**14.3.3 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<sub>MCK</sub>), specify whether the serial clock (f<sub>SCK</sub>) may be input or not, set a start trigger, an operation mode (CSI, UART, or simplified I<sup>2</sup>C), 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 14-7. Format of Serial Mode Register mn (SMRmn) (1/2)**

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), After reset: 0020H R/W  
 F0150H, F0151H (SMR10) to F0156H, F0157H (SMR13)

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 <sup>Note</sup>	0	SIS mn0 <small>Note</small>	1	0	0	MD mn2	MD mn1	MD mn0

CKS mn	Selection of operation clock (f <sub>MCK</sub> ) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock (f <sub>MCK</sub> ) 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 <sub>TCLK</sub> ) is generated.	

CCS mn	Selection of transfer clock (f <sub>TCLK</sub> ) of channel n
0	Divided operation clock f <sub>MCK</sub> specified by the CKSmn bit
1	Clock input f <sub>SCK</sub> from the SCKp pin (slave transfer in CSI mode)
Transfer clock f <sub>TCLK</sub> 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 <sub>MCK</sub> ) is set by the higher 7 bits of the SDRmn register.	

STS mn	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I <sup>2</sup> C).
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, SMR03, SMR11, and SMR13 registers 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, SMR02, SMR10, or SMR12 register) to “0”. Be sure to set bit 5 to “1”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10), q: UART number (q = 0 to 3), r: IIC number (r = 00, 10)

**Figure 14-7. Format of Serial Mode Register mn (SMRmn) (2/2)**

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), After reset: 0020H R/W  
 F0150H, F0151H (SMR10) to F0156H, F0157H (SMR13)

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 <sup>Note</sup>	0	SIS mn0 <sup>Note</sup>	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0	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	CSI mode	
0	1	UART mode	
1	0	Simplified I <sup>2</sup> C 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, SMR03, SMR11, and SMR13 registers 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, SMR02, SMR10, or SMR12 register) to “0”. Be sure to set bit 5 to “1”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10),  
 q: UART number (q = 0 to 3), r: IIC number (r = 00, 10)

**14.3.4 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.








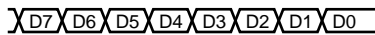




Reset signal generation sets the SCRmn register to 0087H.

**Figure 14-8. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)**

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), After reset: 0087H R/W  
 F0158H, F0159H (SCR10) to F015EH, F015FH (SCR13)

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 <sup>Note 1</sup>	SLC mn0	0	1	DLSm n1 <sup>Note 2</sup>	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 CSI mode	Type
0	0	SCKp  SOp  Slp input timing 	1
0	1	SCKp  SOp  Slp input timing 	2
1	0	SCKp  SOp  Slp input timing 	3
1	1	SCKp  SOp  Slp input timing 	4

Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I<sup>2</sup>C mode.

EOC mn	Selection of masking of error interrupt signal (INTSREx (x = 0 to 3))
0	Masks error interrupt INTSREx (INTSRx is not masked).
1	Enables generation of error interrupt INTSREx (INTSRx is masked if an error occurs).
Set EOCmn = 0 in the CSI mode, simplified I <sup>2</sup> C mode, and during UART transmission <sup>Note 3</sup> .	

- Notes**
1. The SCR00, SCR02, SCR10, and SCR12 registers only.
  2. The SCR00, SCR01, SCR10 and SCR11 registers only. Others are fixed to 1.
  3. When using CSImn not with EOCmn = 0, error interrupt INTSREn may be generated.

**Caution** Be sure to clear bits 3, 6, and 11 to “0” (Also clear bit 5 of the SCR01, SCR03, SCR11, or SCR13 register to 0). Be sure to set bit 2 to “1”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10)

**Figure 14-8. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)**

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), After reset: 0087H R/W  
 F0158H, F0159H (SCR10) to F015EH, F015FH (SCR13)

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 <sup>Note 1</sup>	SLC mn0	0	1	DLSm n1 <sup>Note 2</sup>	DLS mn0

PTC mn1	PTC mn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	
0	1	Outputs 0 parity <sup>Note 3</sup> .	
1	0	Outputs even parity.	
1	1	Outputs odd parity.	
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the CSI mode and simplified I <sup>2</sup> C mode.			

DIR mn	Selection of data transfer sequence in 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 I <sup>2</sup> C mode.		

SLCm n1 <sup>Note 1</sup>	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 I <sup>2</sup> C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the CSI mode. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) or 2 bits (SLCmn1, SLCmn0 = 1, 0) during UART transmission.				

DLSm n1 <sup>Note 2</sup>	DLS mn0	Setting of data length in 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 I <sup>2</sup> C mode.				

- Notes**
1. The SCR00, SCR02, SCR10, and SCR12 registers only.
  2. The SCR00, SCR01, SCR10 and SCR11 registers only. Others are fixed to 1.
  3. 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, SCR03, SCR11, or SCR13 register to 0). Be sure to set bit 2 to “1”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10)

### 14.3.5 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, SDR10, SDR11 or bits 7 to 0 (lower 8 bits) of SDR02, SDR03, SDR12 and SDR13 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 ( $f_{MCK}$ ,  $f_{SCK}$ ).

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 (higher 7 bits) of SDR00, SDR01, SDR10, and SDR11 to 0000000B. The input clock  $f_{SCK}$  (slave transfer in CSI mode) from the SCKp pin is used as the transfer clock.

The lower 8/9 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 8/9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8/9 bits.

The SDRmn register can be read or written in 16-bit units.

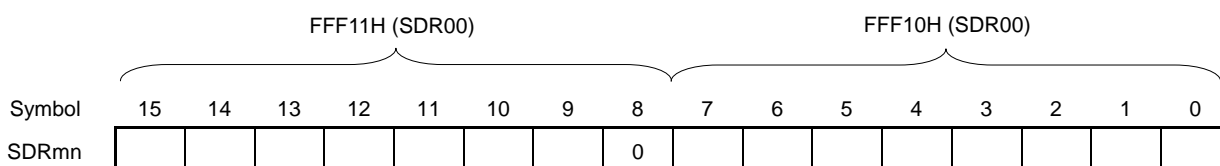
However, the higher 7 bits can be written or read only when the operation is stopped ( $SE_{mn} = 0$ ). During operation ( $SE_{mn} = 1$ ), a value is written only to the lower 8/9 bits of the SDRmn register. When the SDRmn register is read during operation, 0 is always read.

Reset signal generation clears the SDRmn register to 0000H.

**Figure 14-9. Format of Serial Data Register mn (SDRmn)**

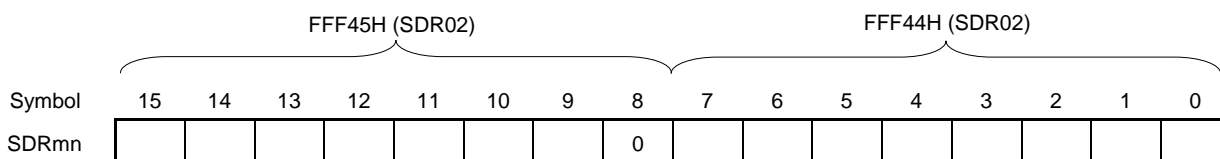
Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01),  
 FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)

After reset: 0000H R/W



Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),  
 FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)

After reset: 0000H R/W



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 and Remarks are listed on the next page.)

&lt;R&gt;

- Cautions**
1. Be sure to clear bit 8 of the SDR02, SDR03, SDR12, and SDR13 registers to “0”.
  2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.
  3. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I<sup>2</sup>C is used. Set SDRmn[15:9] to 0000001B or greater.
  4. 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 8/9 bits of the SDRmn register, see 14.2 Configuration of Serial Array Unit.
  2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

**14.3.6 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 14-10. Format of Serial Flag Clear Trigger Register mn (SIRmn)**

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03), After reset: 0000H R/W  
 F0148H, F0149H (SIR10) to F014EH, F014FH (SIR13)

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	FECT mn <sup>Note</sup>	PEC Tmn	OVC Tmn

FEC Tmn	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PEC Tmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVC Tmn	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, SIR03, SIR11, and SIR13 registers only.

**Caution** Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00, SIR02, SIR10, or SIR12 register) to “0”.

- Remarks**
1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
  2. When the SIRmn register is read, 0000H is always read.

**14.3.7 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 14-11. Format of Serial Status Register mn (SSRmn) (1/2)**

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), After reset: 0000H Rs  
 F0140H, F0141H (SSR10) to F0146H, F0147H (SSR13)

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	FEF mn <sup>Note</sup>	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> <ul style="list-style-type: none"> <li>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).</li> <li>Communication ends.</li> </ul> <Set condition> <ul style="list-style-type: none"> <li>Communication starts.</li> </ul>	

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> <ul style="list-style-type: none"> <li>Transferring transmit data from the SDRmn register to the shift register ends during transmission.</li> <li>Reading receive data from the SDRmn register ends during reception.</li> <li>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).</li> </ul> <Set conditions> <ul style="list-style-type: none"> <li>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).</li> <li>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).</li> <li>A reception error occurs.</li> </ul>	

**Note** The SSR01, SSR03, SSR11, and SSR13 registers only.

**Caution** 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.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)



**Figure 14-11. Format of Serial Status Register mn (SSRmn) (2/2)**

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), After reset: 0000H R  
 F0140H, F0141H (SSR10) to F0146H, F0147H (SSR13)

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	FEF mn <sup>Note</sup>	PEF mn	OVF mn

FEF mn	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear condition> <ul style="list-style-type: none"> <li>• 1 is written to the FECTmn bit of the SIRmn register.</li> </ul> <Set condition> <ul style="list-style-type: none"> <li>• A stop bit is not detected when UART reception ends.</li> </ul>	

PEF mn	Parity/ACK error detection flag of channel n
0	No error occurs.
1	Parity error occurs (during UART reception) or ACK is not detected (during I <sup>2</sup> C transmission).
<Clear condition> <ul style="list-style-type: none"> <li>• 1 is written to the PECTmn bit of the SIRmn register.</li> </ul> <Set condition> <ul style="list-style-type: none"> <li>• The parity of the transmit data and the parity bit do not match when UART reception ends (parity error).</li> <li>• No ACK signal is returned from the slave channel at the ACK reception timing during I<sup>2</sup>C transmission (ACK is not detected).</li> </ul>	

OVF mn	Overflow error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> <ul style="list-style-type: none"> <li>• 1 is written to the OVCTmn bit of the SIRmn register.</li> </ul> <Set condition> <ul style="list-style-type: none"> <li>• 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).</li> <li>• Transmit data is not ready for slave transmission or transmission and reception in CSI mode.</li> </ul>	

**Note** The SSR01, SSR03, SSR11, and SSR13 registers only.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

**14.3.8 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 a 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears the SSm register to 0000H.

**Figure 14-12. Format of Serial Channel Start Register m (SSm)**

Address: F0122H, F0123H (SS0) 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	SS03	SS02	SS01	SS00

Address: F0162H, F0163H (SS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13	SS12	SS11	SS10

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEmn bit to 1 and enters the communication wait status <sup>Note</sup> .

**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 4 of the SS0 register, bits 15 to 2 of the SS1 register for 64-pin products and bits 15 to 4 of the SS1 register for 80-pin products 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<sub>MCK</sub> clocks have elapsed.

- Remarks**
1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
  2. When the SSm register is read, 0000H is always read.

**14.3.9 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 14-13. Format of Serial Channel Stop Register m (STm)**

Address: F0124H, F0125H (ST0) After reset: 0000H 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	ST03	ST02	ST01	ST00

Address: F0164H, F0165H (ST1) After reset: 0000H W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST1	0	0	0	0	0	0	0	0	0	0	0	0	ST13	ST12	ST11	ST10

STm n	Operation stop trigger of channel n															
0	No trigger operation															
1	Clears the SEmn bit to 0 and stops the communication operation <sup>Note</sup> .															

**Note** Holding status value of the control register and shift register, the SCKmn and SOMn pins, and FEFmn, PEFmn, OVFmn flags.

**Caution** Be sure to clear bits 15 to 4 of the ST0 register, bits 15 to 2 of the ST1 register for 64-pin products and bits 15 to 4 of the ST1 register for 80-pin products to “0”.

**Remarks** 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)  
 2. When the STm register is read, 0000H is always read.

**14.3.10 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 CKOm<sub>n</sub> 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 CKOm<sub>n</sub> 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 14-14. Format of Serial Channel Enable Status Register m (SEm)**

Address: F0120H, F0121H (SE0) 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	SE03	SE02	SE01	SE00

Address: F0160H, F0161H (SE1) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE1	0	0	0	0	0	0	0	0	0	0	0	0	SE13	SE12	SE11	SE10

SEm <sub>n</sub>	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

**14.3.11 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 14-15. Format of Serial Output Enable Register m (SOEm)**

Address: F012AH, F012BH (SOE0) 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	SOE 02	0	SOE 00

Address: F016AH, F016BH (SOE1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE 12	0	SOE 10

SOE mn	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 3 and 1 of the SOE0 register, bits 15 to 1 of the SOE1 register for 64-pin products and bits 15 to 3 and 1 of the SOE1 register for 80-pin products to “0”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

**14.3.12 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 0F0FH.

**Figure 14-16. Format of Serial Output Register m (SOM)**

Address: F0128H, F0129H (SO0) After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO 02	1	CKO 00	0	0	0	0	1	SO 02	1	SO 00

Address: F0168H, F0169H (SO1) After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	1	0	0	0	0	1	SO 12	1	SO 10

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 12 and 7 to 4 of the SO0 register to “0”. And be sure to set bits 11, 9, 3 and 1 to “1”.

Be sure to clear bits 15 to 12 and 7 to 4 of the SO1 register to “0”. And be sure to set bits 11 to 8 and 3 to 1 for 64-pin products and bits 11 to 8, 3 and 1 for 80-pin products to “1”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

**14.3.13 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 CSI mode and simplifies I<sup>2</sup>C 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 14-17. Format of Serial Output Level Register m (SOLm)**

Address: F0134H, F0135H (SOL0) 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	SOL 02	0	SOL 00

Address: F0174H, F0175H (SOL1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 12	0	SOL 10

SOL mn	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.

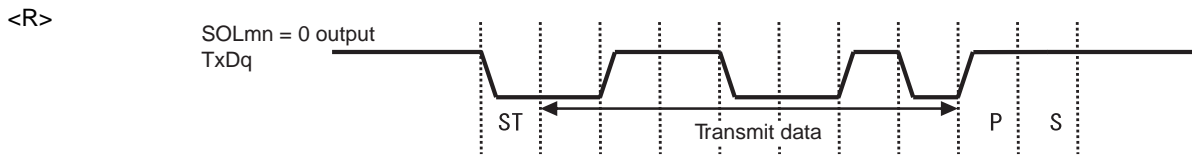
**Caution** Be sure to clear bits 15 to 3, and 1 of the SOL0 register, bits 15 to 1 of the SOL1 register for 64-pin products, and 15 to 3, and 1 of the SOL1 register for 80-pin products to “0”.

(Remark is listed on the next page.)

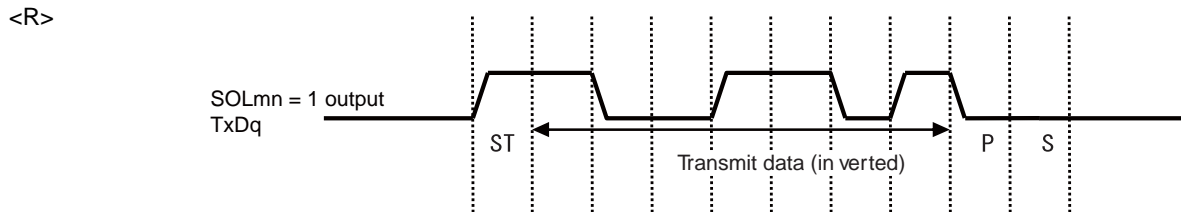
Figure 14-18 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 14-18. Examples of Reverse Transmit Data

(a) Non-reverse Output (SOLmn = 0)



(b) Reverse Output (SOLmn = 1)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)



**14.3.14 Serial standby control register m (SSCm)**

The SSC0 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI00 or UART0 serial data.

The SSC1 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving UART2 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
- When using UART0, UART2: 9600 bps

**Figure 14-19. Format of Serial Standby Control Register m (SSCm)**

Address: F0138H (SSC0), F0178H (SSC1) 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	SS ECm	SWC m

SS ECm	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/INTSRE2).
1	Stop the generation of error interrupts (INTSRE0/INTSRE2).
<ul style="list-style-type: none"> <li>• The SSECm bit can be set to 1 or 0 only when both the SWCm and EOCmn bits are set to 1 during UART reception in the SNOOZE mode. In other cases, clear the SSECm bit to 0.</li> <li>• Setting SSECm, SWCm = 1, 0 is prohibited.</li> </ul>	

SWC m	Setting of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.
<ul style="list-style-type: none"> <li>• 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).</li> <li>• 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<sub>CLK</sub>). If any other clock is selected, specifying this mode is prohibited.</li> <li>• Even when using SNOOZE mode, be sure to set the SWCm bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode.</li> </ul> <p>Also, be sure to change the SWCm bit to 0 after returning from STOP mode to normal operation mode.</p>	

**Figure 14-20. Interrupt in UART Reception Operation in SNOOZE Mode**

EOCmn Bit	SSECm Bit	Reception Ended Successfully	Reception Ended in an Error
0	0	INTSRx is generated.	INTSRx is generated.
0	1	INTSRx is generated.	INTSRx is generated.
1	0	INTSRx is generated.	INTSREx is generated.
1	1	INTSRx is generated.	No interrupt is generated.

**14.3.15 Input switch control register (ISC)**

The ISC1 and ISC0 bits of the ISC register are used to realize a LIN-bus communication operation by UART0 in coordination with an external interrupt and the timer array unit.

When bit 0 is set to 1, the input signal of the serial data input (RxD0) pin is selected as an external interrupt (INTP0) that can be used to detect a wakeup signal.

When bit 1 is set to 1, the input signal of the serial data input (RxD0) pin is selected as a timer input, so that wake up signal can be detected, the low width of the break field, and the pulse width of the sync field can be measured by the timer.

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

Reset signal generation clears the ISC register to 00H.

**Figure 14-21. Format of Input Switch Control Register (ISC)**

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of the RxD0 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD0 pin as an external interrupt (wakeup signal detection).

- Caution**
1. Be sure to clear bits 7 to 2 to “0”.
  2. 16-bit timer KB20 cannot be used together with the LIN-bus functions. When using 16-bit timer KB20, set the ISC register to its initial value (00H).

**14.3.16 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 CSI or simplified I<sup>2</sup>C 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<sub>MCK</sub>) 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<sub>MCK</sub>) 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.

**Figure 14-22. 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	SNFEN30	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN30	Use of noise filter of RxD2 pin (RxD2/SDA20/SI20/P14)
0	Noise filter OFF
1	Noise filter ON
Set SNFEN30 to 1 to use the RxD3 pin. Clear SNFEN30 to 0 to use the other than RxD3 pin.	

SNFEN20	Use of noise filter of RxD2 pin (RxD2/SDA20/SI20/P14)
0	Noise filter OFF
1	Noise filter ON
Set SNFEN20 to 1 to use the RxD2 pin. Clear SNFEN20 to 0 to use the other than RxD2 pin.	

SNFEN10	Use of noise filter of RxD1 pin (RXD1/ANI16/SI10/SDA10/P03)
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN10 bit to 1 to use the RxD1 pin. Clear the SNFEN10 bit to 0 to use the other than RxD1 pin.	

SNFEN00	Use of noise filter of RxD0 pin (RXD0/TOOLRXD/SDA00/SI00/P11)
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 5, 3, and 1 for 64-pin products and bits 7, 5, 3, and 1 for 80-pin products to “0”.

### 14.3.17 Registers controlling port functions of serial I/O pins

When using the serial array unit set 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), port mode control register (PMCxx)).

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)**.

When using a port pin with a multiplexed serial data or serial clock output function (e.g. P07/SO10/TxD1/SEG50) for serial data or serial clock output, set the corresponding bits in the port mode control register (PMCxx) 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: When P07/SO10/TxD1/SEG50 is to be used for serial data output

Set the PM07 bit of port mode register 0 to 0.

Set the P07 bit of port register 0 to 1.

When using a port pin with a multiplexed serial data or serial clock input function (e.g. P06/SI10/RxD1/SDA10/SEG49) for serial data or serial clock input, set 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 P06/SI10/RxD1/SDA10/SEG49 is to be used for serial data input

Set the PM06 bit of port mode register 0 to 1.

Set the P06 bit of port register 0 to 0 or 1.

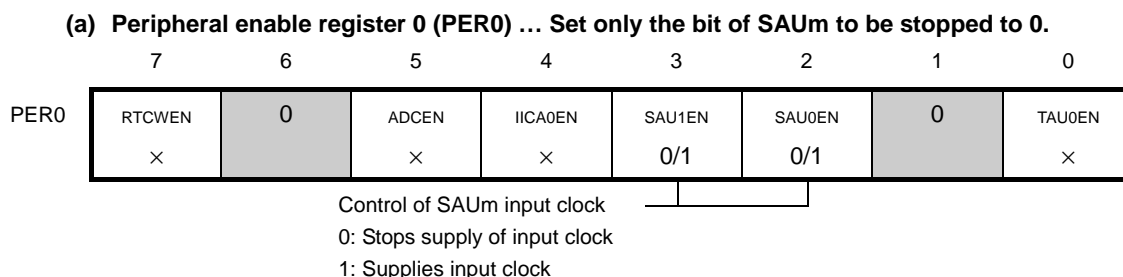
### 14.4 Operation Stop Mode

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.

#### 14.4.1 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 0, set bit 2 (SAU0EN) to 0. To stop the operation of serial array unit 1, set bit 3 (SAU1EN) to 0.

Figure 14-23. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units



**Cautions 1.** If SAUmEN = 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.

- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 0, 1, 3 (PIM0, PIM1, PIM3)
- Port output mode registers 0, 1, 3 (POM0, POM1, POM3)
- Port mode registers 0, 1, 3 (PM0, PM1, PM3)
- Port registers 0, 1, 3 (P0, P1, P3)

**2.** Be sure to clear bits 1, 6 to 0.

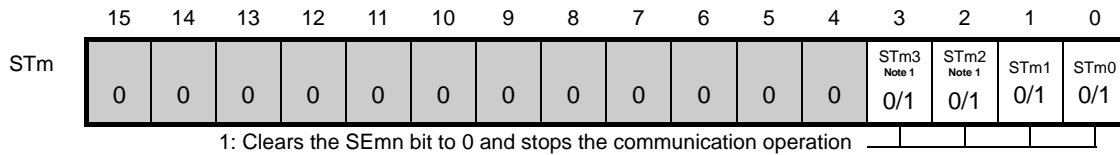
**Remark** x: Bits not used with serial array units (depending on the settings of other peripheral functions)  
 0/1: Set to 0 or 1 depending on the usage of the user

14.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

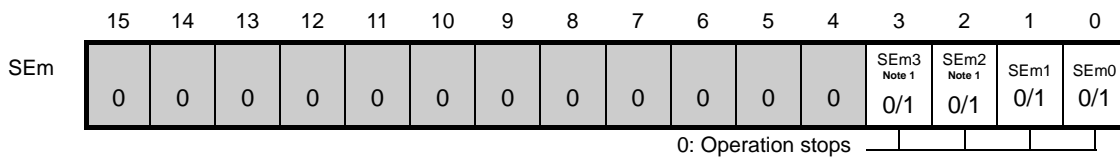
Figure 14-24. Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm) ... This register is a trigger register that is used to enable stopping communication/count by each channel.



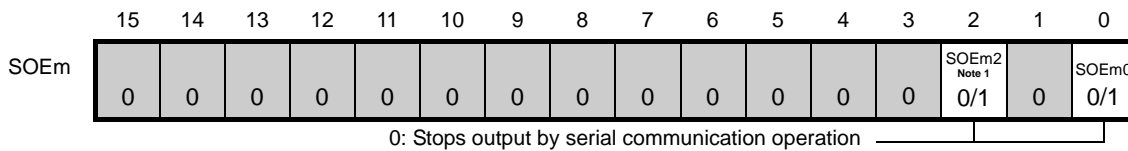
\* Because the ST<sub>m</sub>n bit is a trigger bit, it is cleared immediately when SE<sub>m</sub>n = 0.

(b) Serial Channel Enable Status Register m (SEm) ... This register indicates whether data transmission/reception operation of each channel is enabled or stopped.



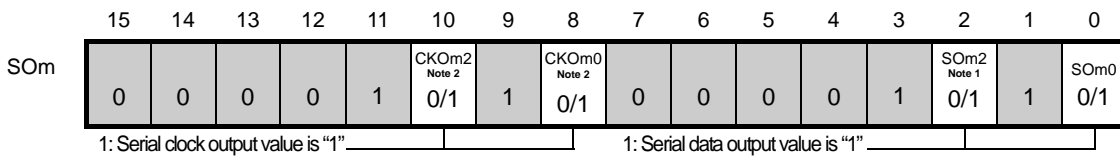
\* The SEm register is a read-only status register, whose operation is stopped by using the STm register. With a channel whose operation is stopped, the value of the CKO<sub>m</sub>n bit of the SOm register can be set by software.

(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.



\* For channel n, whose serial output is stopped, the SO<sub>m</sub>n 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.



\* When using pins corresponding to each channel as port function pins, set the corresponding CKO<sub>m</sub>n, SO<sub>m</sub>n bits to "1".

- Notes**
1. When Serial array unit 1, 80-pin products only.
  2. Serial array unit 0 only.

- Remarks**
1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
  2. : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

### 14.5 Operation of 3-Wire Serial I/O (CSI00, CSI10) 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<sup>Note</sup>

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.

**Note** Use the clocks within a range satisfying the SCK cycle time ( $t_{KCY}$ ) characteristics. For details, see **CHAPTER 32 ELECTRICAL SPECIFICATIONS ( $T_A = -40$  to  $+85^\circ\text{C}$ )** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS ( $T_A = -40$  to  $+105^\circ\text{C}$ )**.

The channels supporting 3-wire serial I/O (CSI00, CSI10) are channels 0 and 2 of SAU0.

- 64-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I <sup>2</sup> C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	UART2	–
	1	–		–

- 80-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I <sup>2</sup> C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	UART2	–
	1	–		–
	2	–	UART3	–
	3	–		–

3-wire serial I/O (CSI00, CSI10) performs the following seven types of communication operations.

- Master transmission (See 14.5.1.)
- Master reception (See 14.5.2.)
- Master transmission/reception (See 14.5.3.)
- Slave transmission (See 14.5.4.)
- Slave reception (See 14.5.5.)
- Slave transmission/reception (See 14.5.6.)
- SNOOZE mode function (See 14.5.7.)



### 14.5.1 Master transmission

Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

3-Wire Serial I/O	CSI00	CSI10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCK00, SO00	SCK10, SO10
Interrupt	INTCSI00	INTCSI10
	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 <sup>Note</sup>	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 <ul style="list-style-type: none"> <li>• DAPmn = 0: Data output starts from the start of the operation of the serial clock.</li> <li>• DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.</li> </ul>	
Clock phase	Selectable by the CKPmn bit of the SCRmn register <ul style="list-style-type: none"> <li>• CKPmn = 0: Non-reverse (data output at the falling edge and data input at the rising edge of SCK)</li> <li>• CKPmn = 1: Reverse (data output at the rising edge and data input at the falling edge of SCK)</li> </ul>	
Data direction	MSB or LSB first	

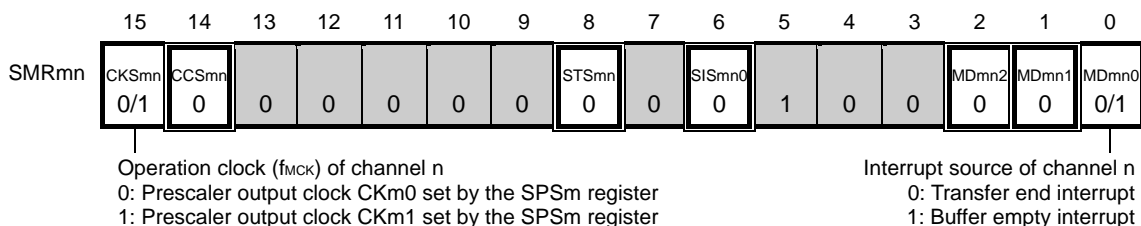
**Note** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)**).

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02

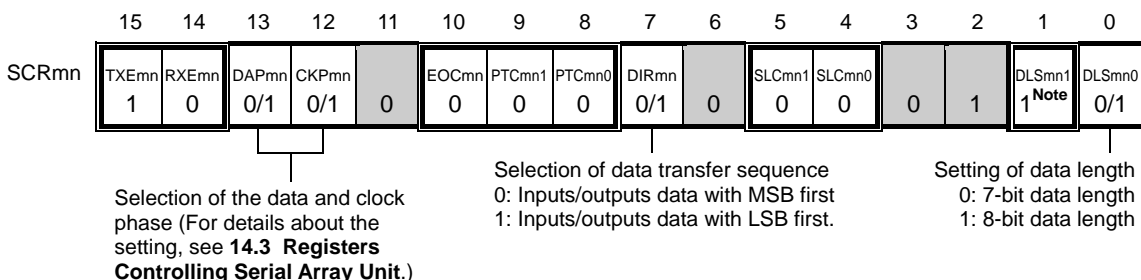
(1) Register setting

Figure 14-25. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI10) (1/2)

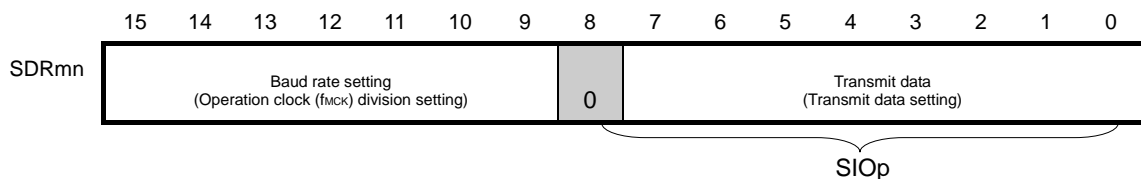
(a) Serial mode register mn (SMRmn)



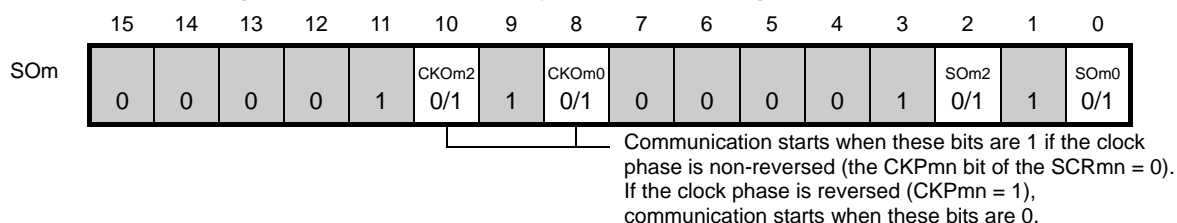
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOM) ... Sets only the bits of the target channel.



**Note** Only provided for the SCR00 register. This bit is fixed to 1 for the other registers.

- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02
  - ◻: Setting is fixed in the 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 14-25. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI10) (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	SOEm2 0/1	0	SOEm0 0/1

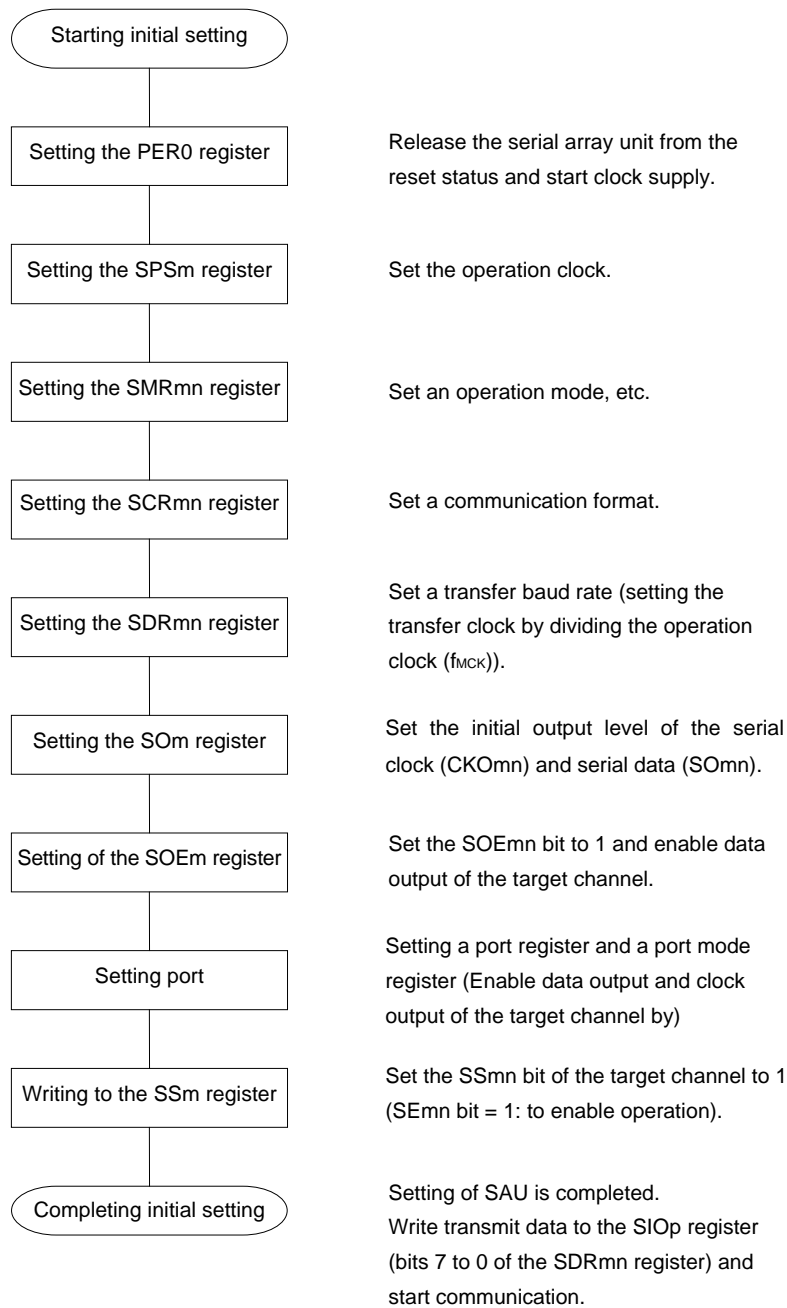
(f) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.

	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	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

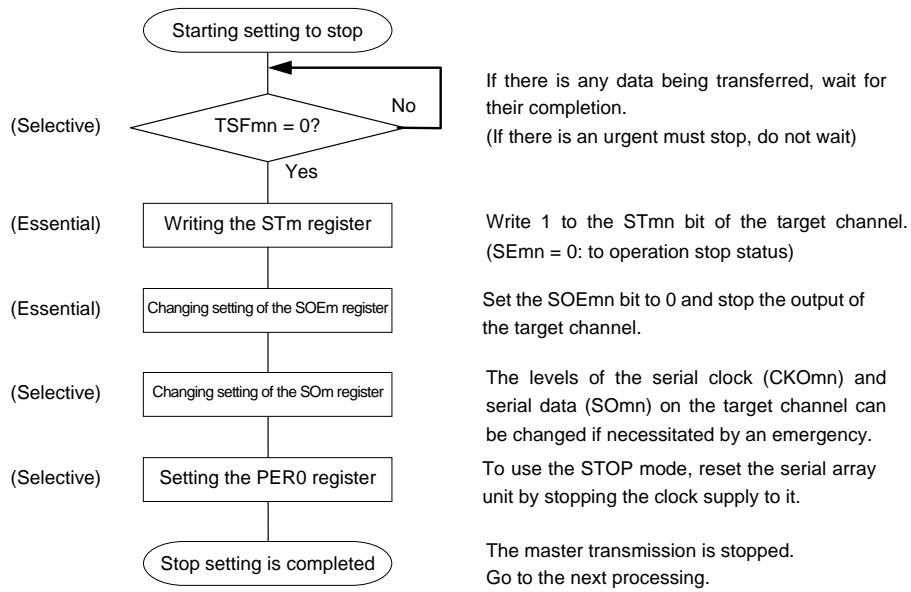
- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02
  - : Setting disabled (set to the initial value)
    - ×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
    - 0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

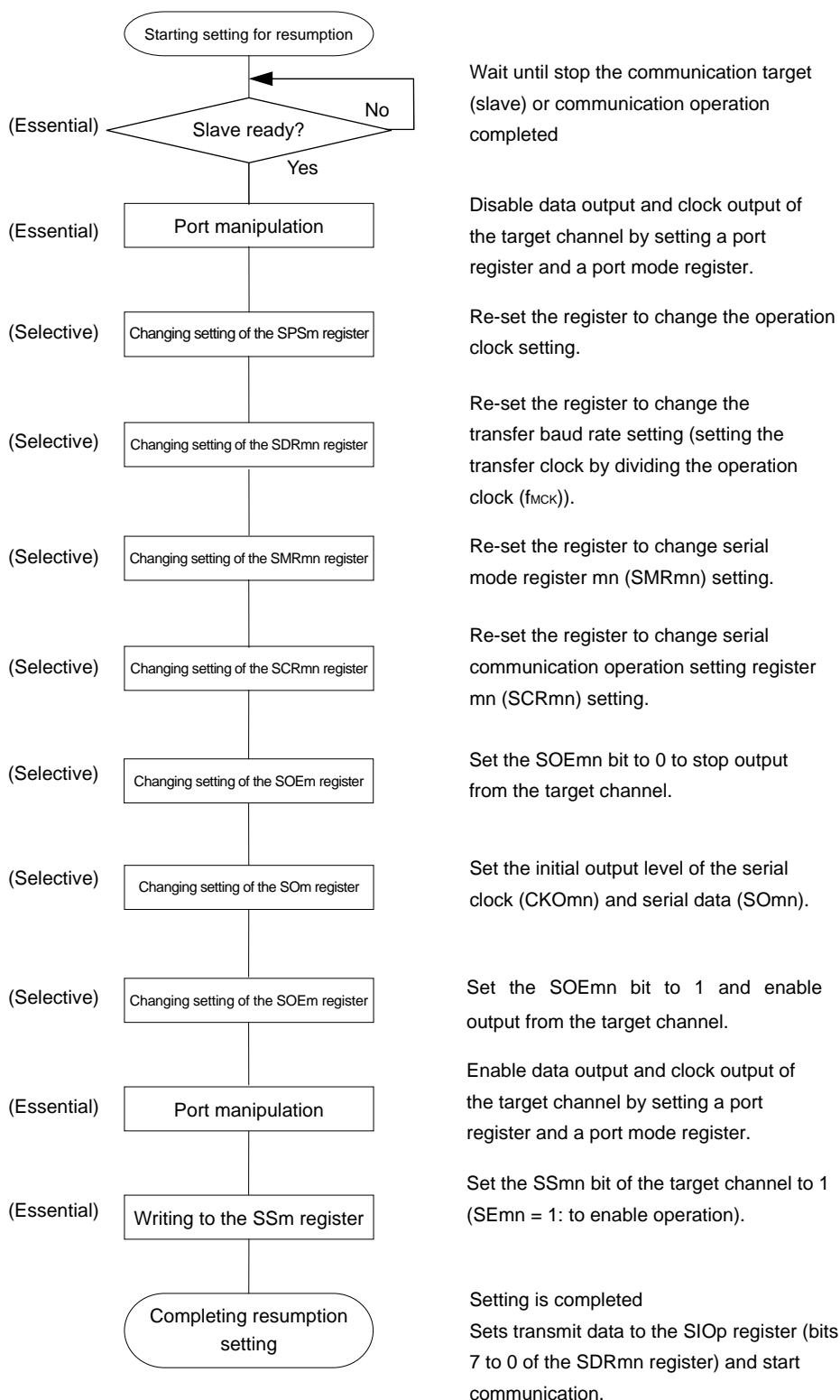
Figure 14-26. Initial Setting Procedure for Master Transmission



**Figure 14-27. Procedure for Stopping Master Transmission**



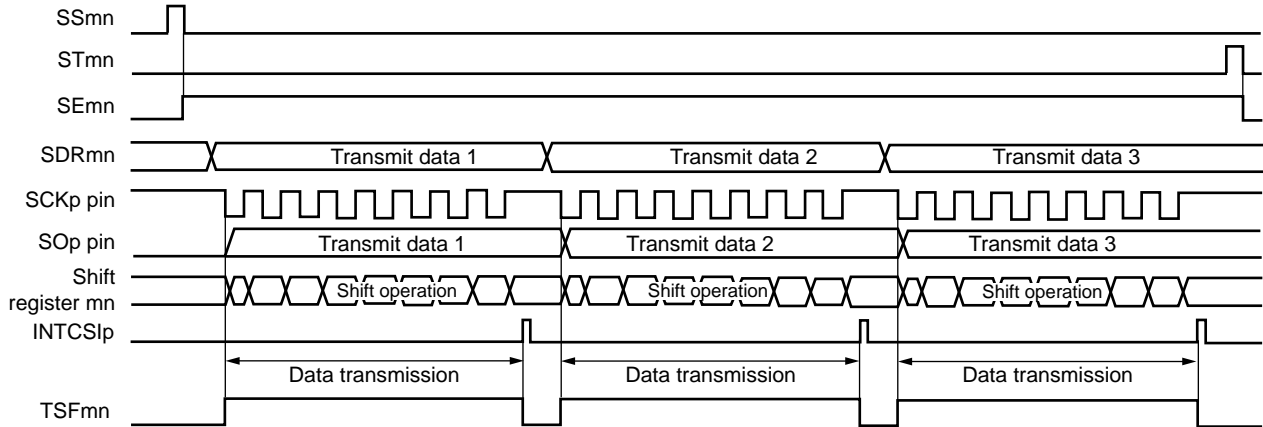
**Figure 14-28. Procedure for Resuming Master Transmission**



**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.

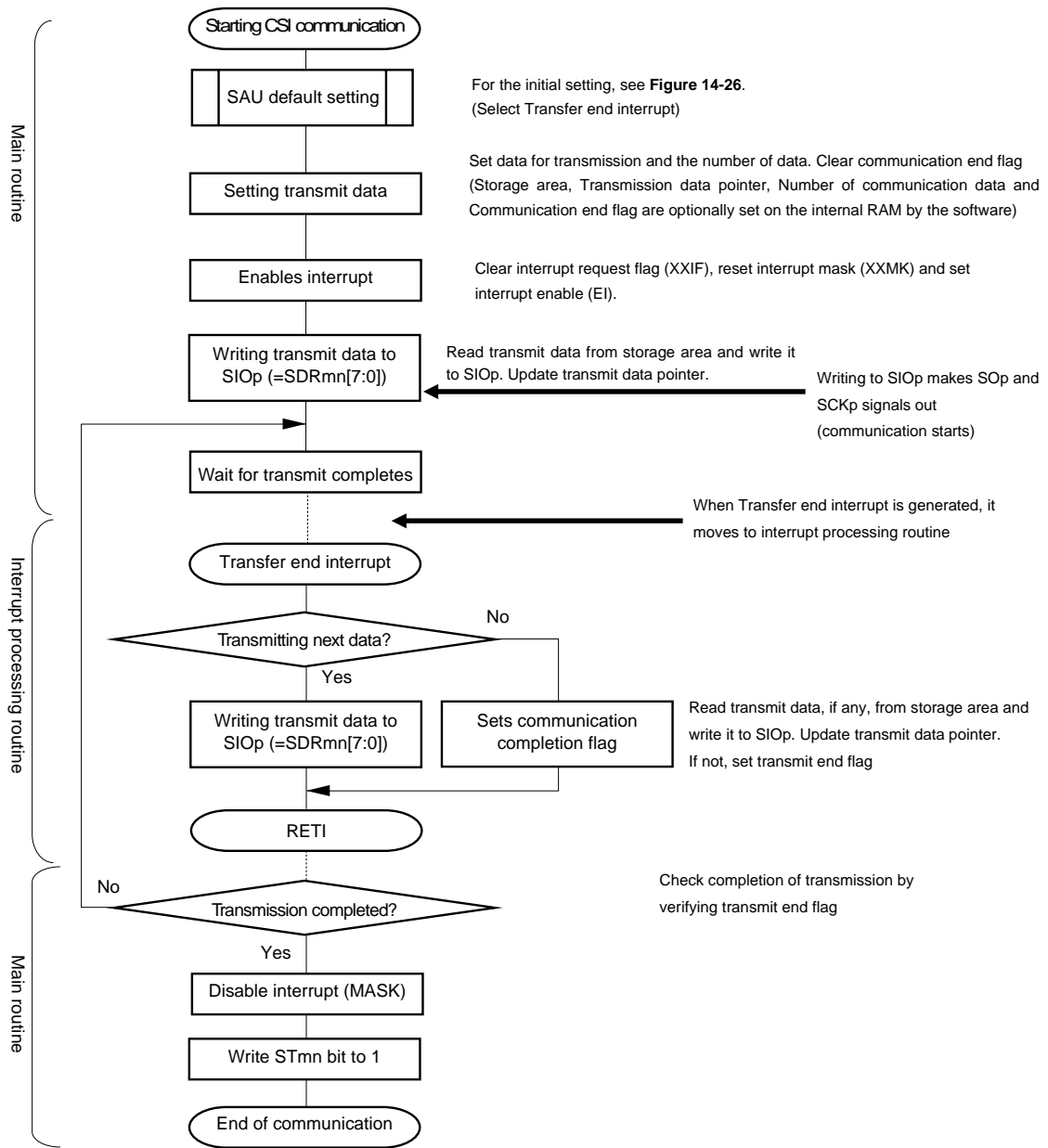
(3) Processing flow (in single-transmission mode)

**Figure 14-29. 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, 2), p: CSI number (p = 00, 10), mn = 00, 02

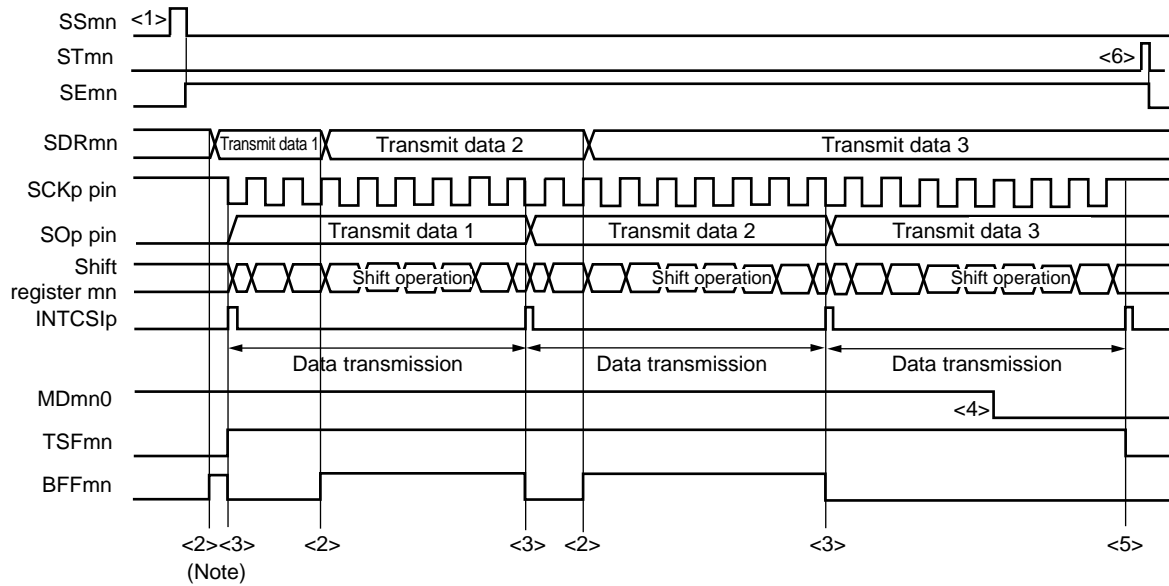
Figure 14-30. Flowchart of Master Transmission (in Single-Transmission Mode)





(4) Processing flow (in continuous transmission mode)

Figure 14-31. Timing Chart of Master Transmission (in Continuous Transmission Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)

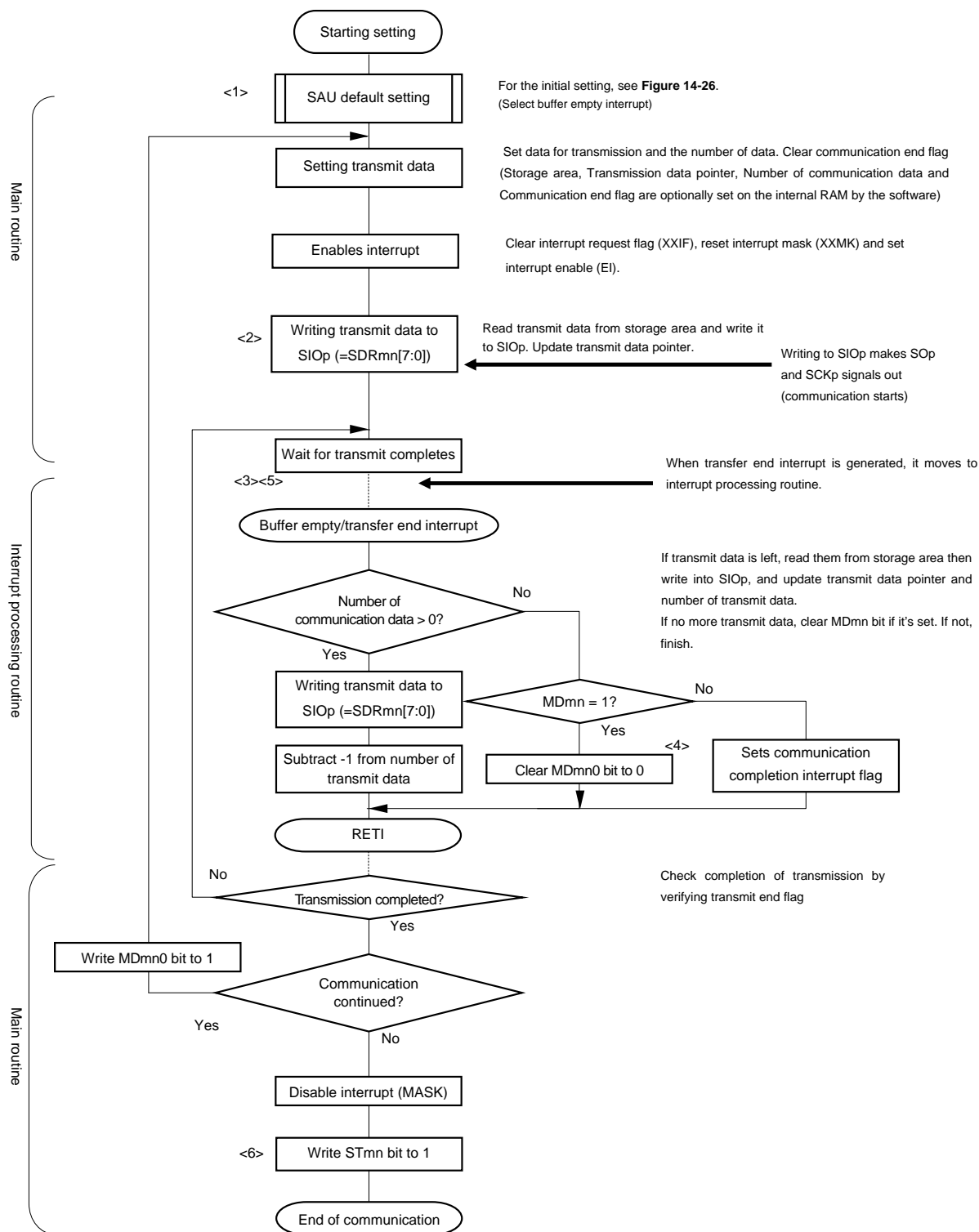


**Note** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

**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, 2), p: CSI number (p = 00, 10), mn = 00, 02

Figure 14-32. Flowchart of Master Transmission (in Continuous Transmission Mode)



**Remark** <1> to <6> in the figure correspond to <1> to <6> in Figure 14-31 Timing Chart of Master Transmission (in Continuous Transmission Mode).

### 14.5.2 Master reception

Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from other device.

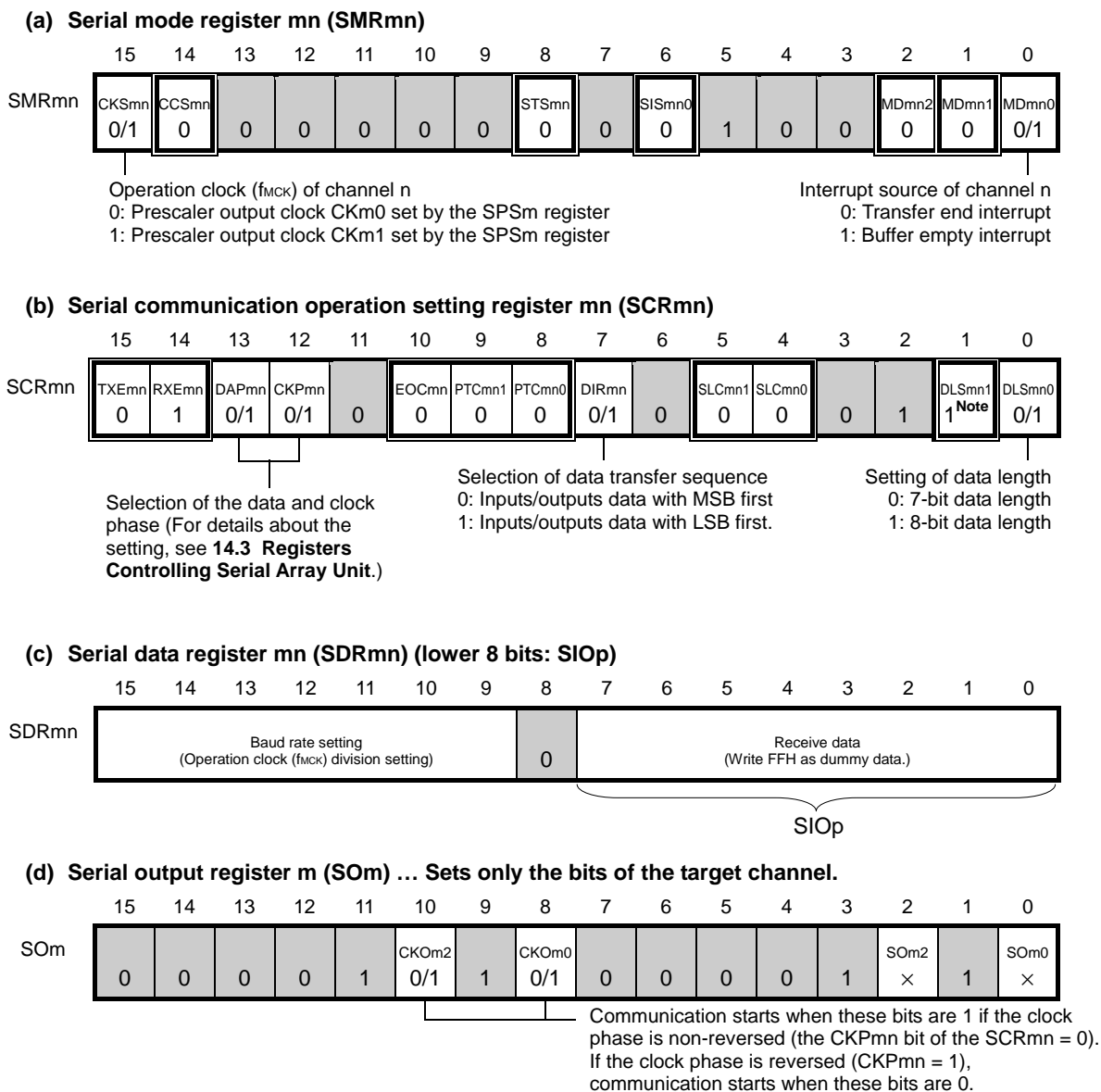
3-Wire Serial I/O	CSI00	CSI10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCK00, SI00	SCK10, SI10
Interrupt	INTCSI00	INTCSI10
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overflow error detection flag (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate <sup>Note</sup>	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 <ul style="list-style-type: none"> <li>• DAPmn = 0: Data input starts from the start of the operation of the serial clock.</li> <li>• DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.</li> </ul>	
Clock phase	Selectable by the CKPmn bit of the SCRmn register <ul style="list-style-type: none"> <li>• CKPmn = 0: Non-reverse</li> <li>• CKPmn = 1: Reverse</li> </ul>	
Data direction	MSB or LSB first	

**Note** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)**).

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02

(1) Register setting

Figure 14-33. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI10) (1/2)



**Note** Only provided for the SCR00 register . This bit is fixed to 1 for the other registers.

- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02
  - : Setting is fixed in the 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 14-33. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI10) (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	SOEm2 ×	0	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	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)
  - : 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 14-34. Initial Setting Procedure for Master Reception

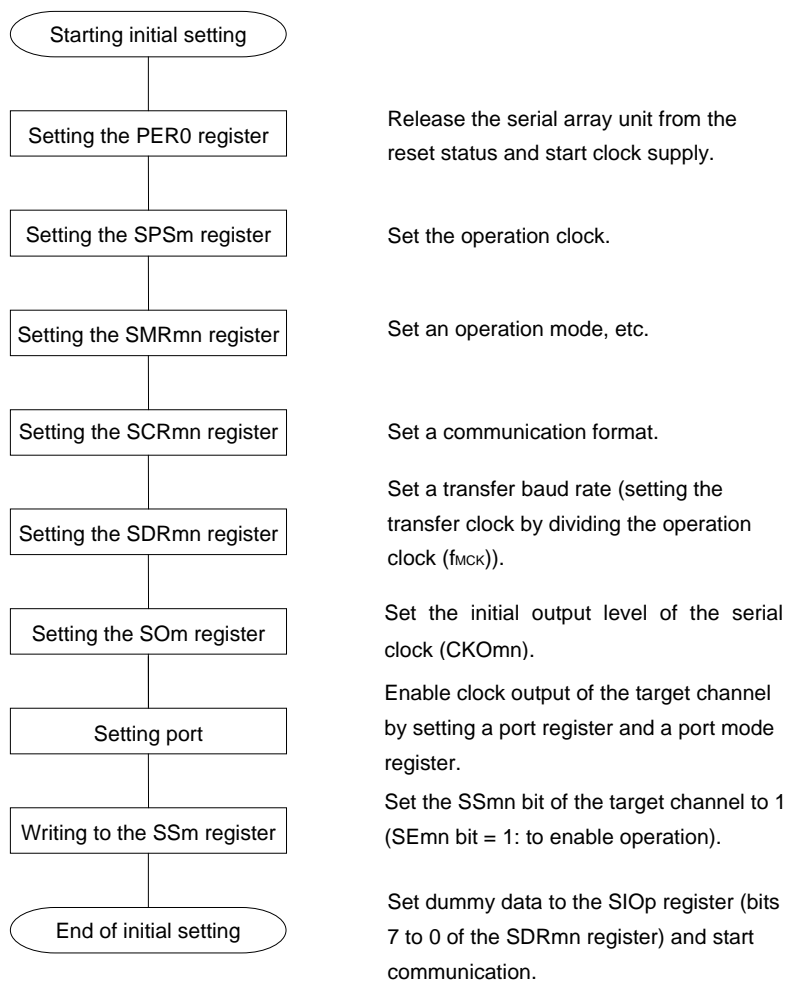
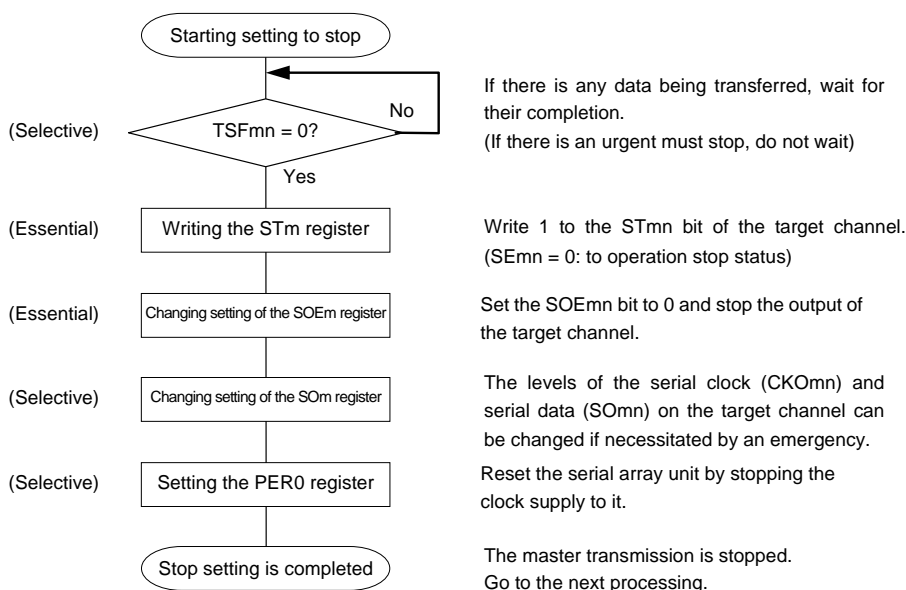
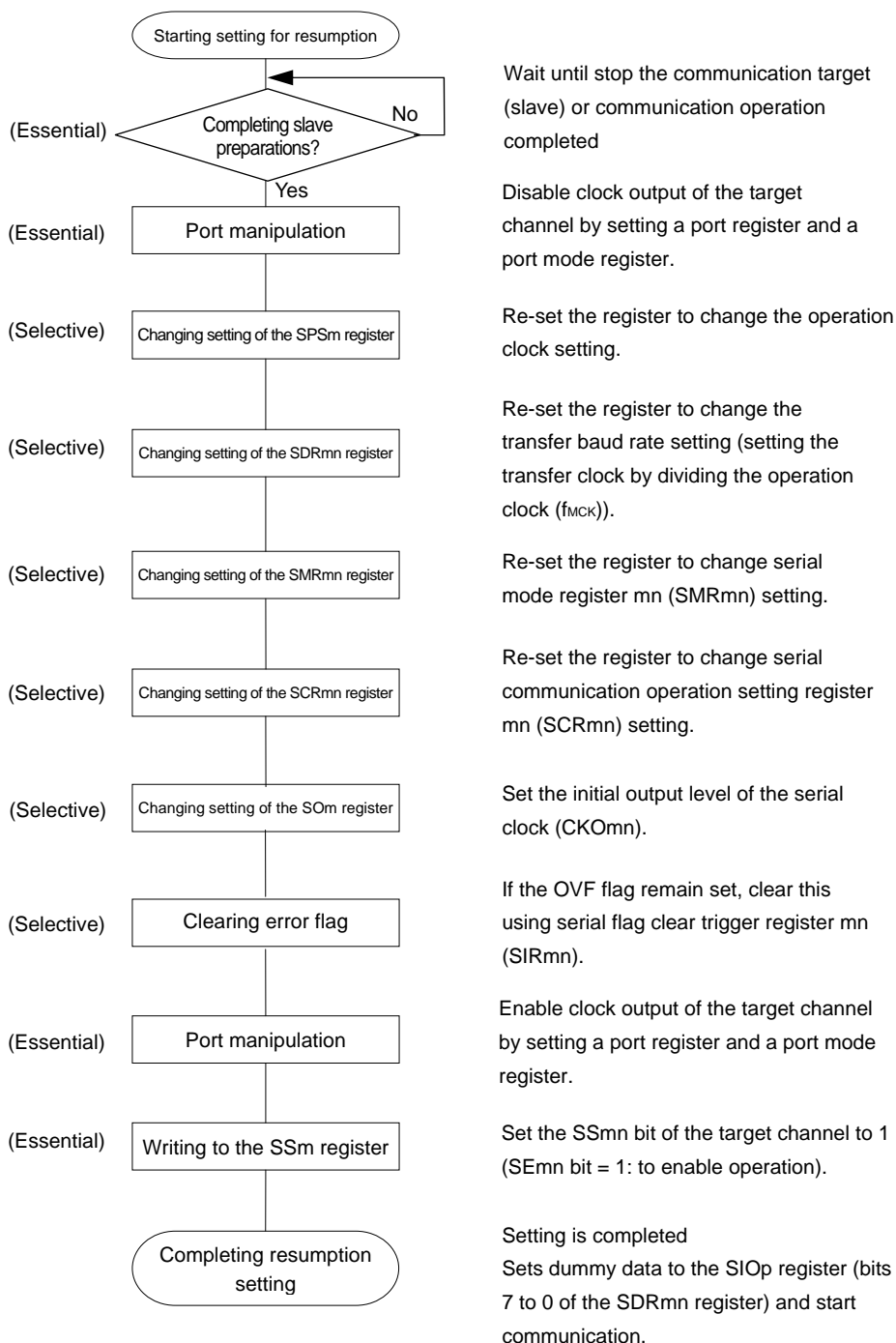


Figure 14-35. Procedure for Stopping Master Reception



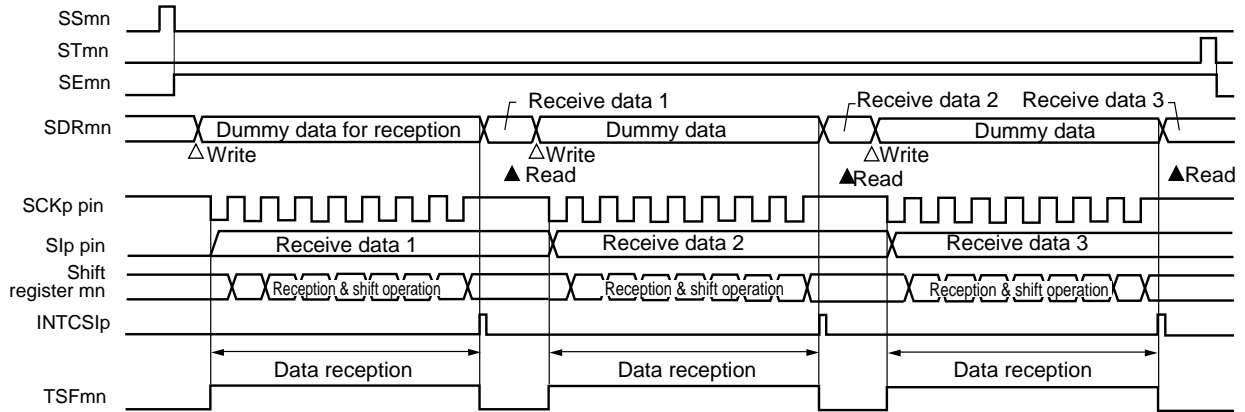
**Figure 14-36. Procedure for Resuming Master Reception**



**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)

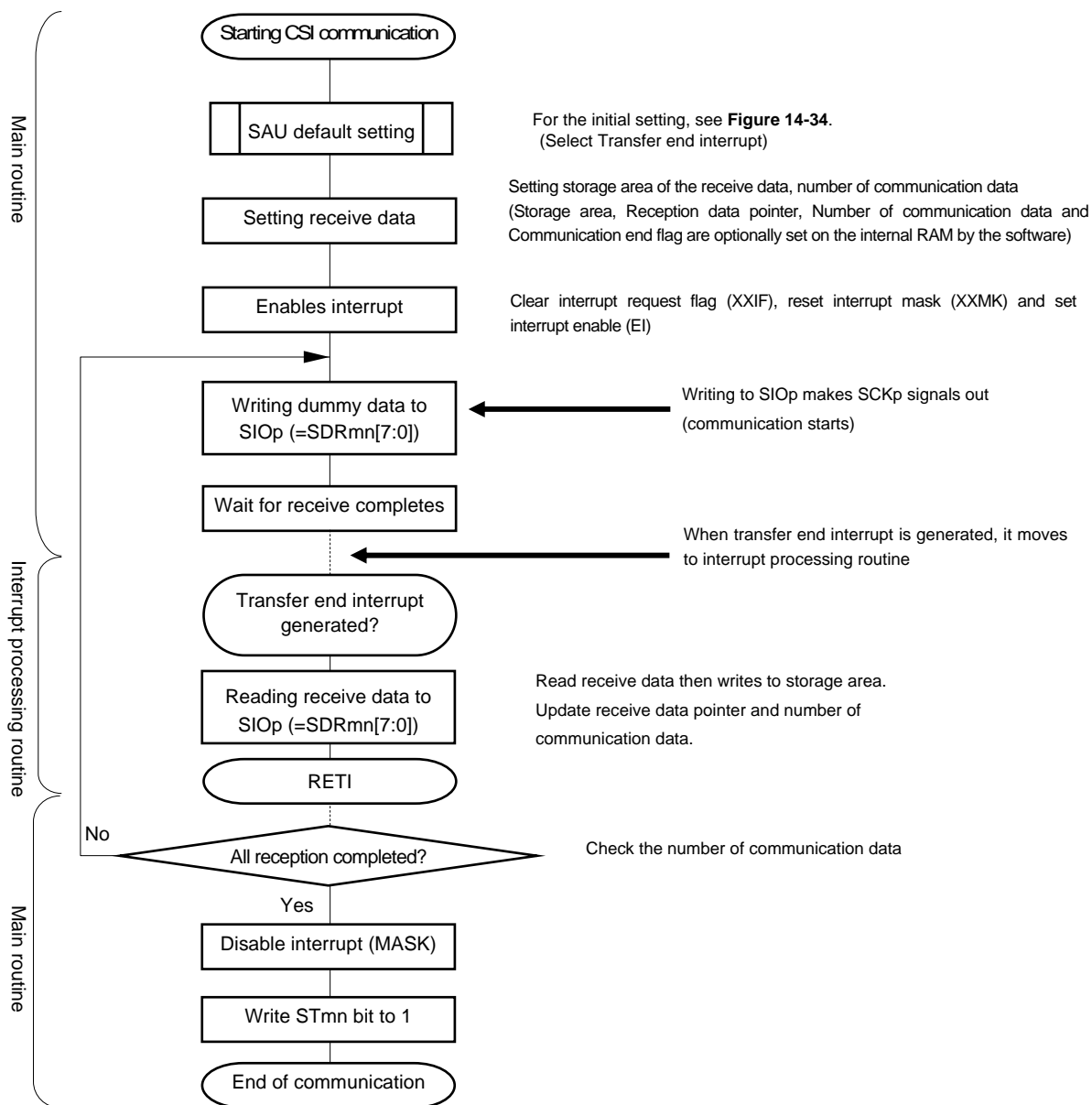
Figure 14-37. Timing Chart of Master Reception (in Single-Reception Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)



**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02

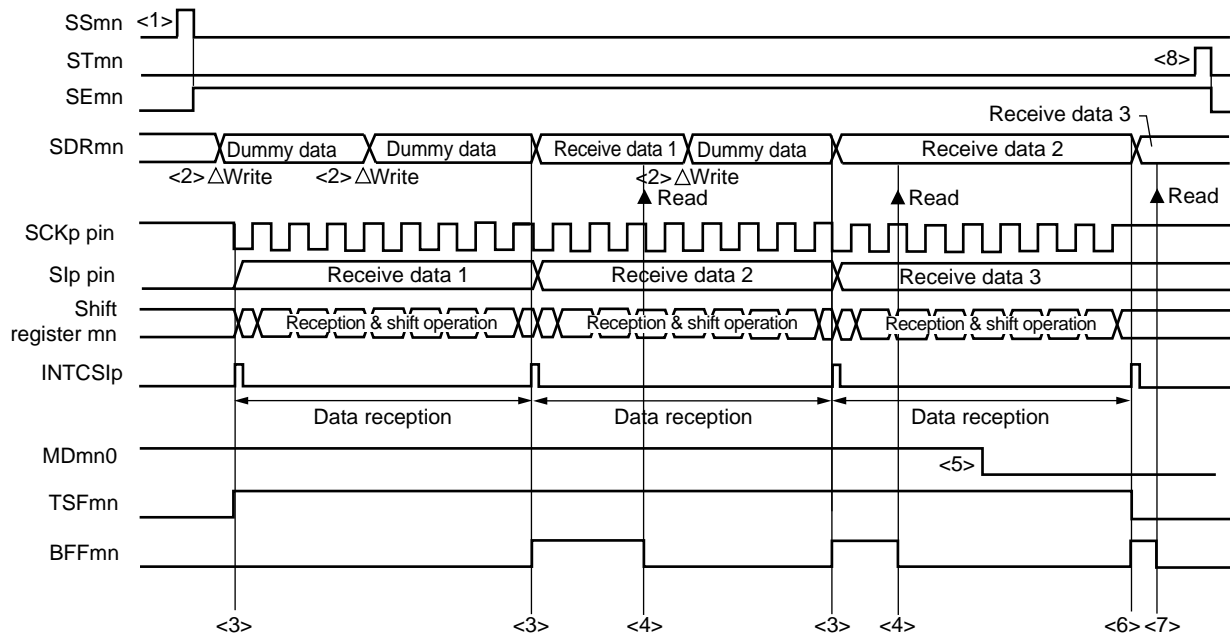


Figure 14-38. Flowchart of Master Reception (in Single-Reception Mode)



(4) Processing flow (in continuous reception mode)

Figure 14-39. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

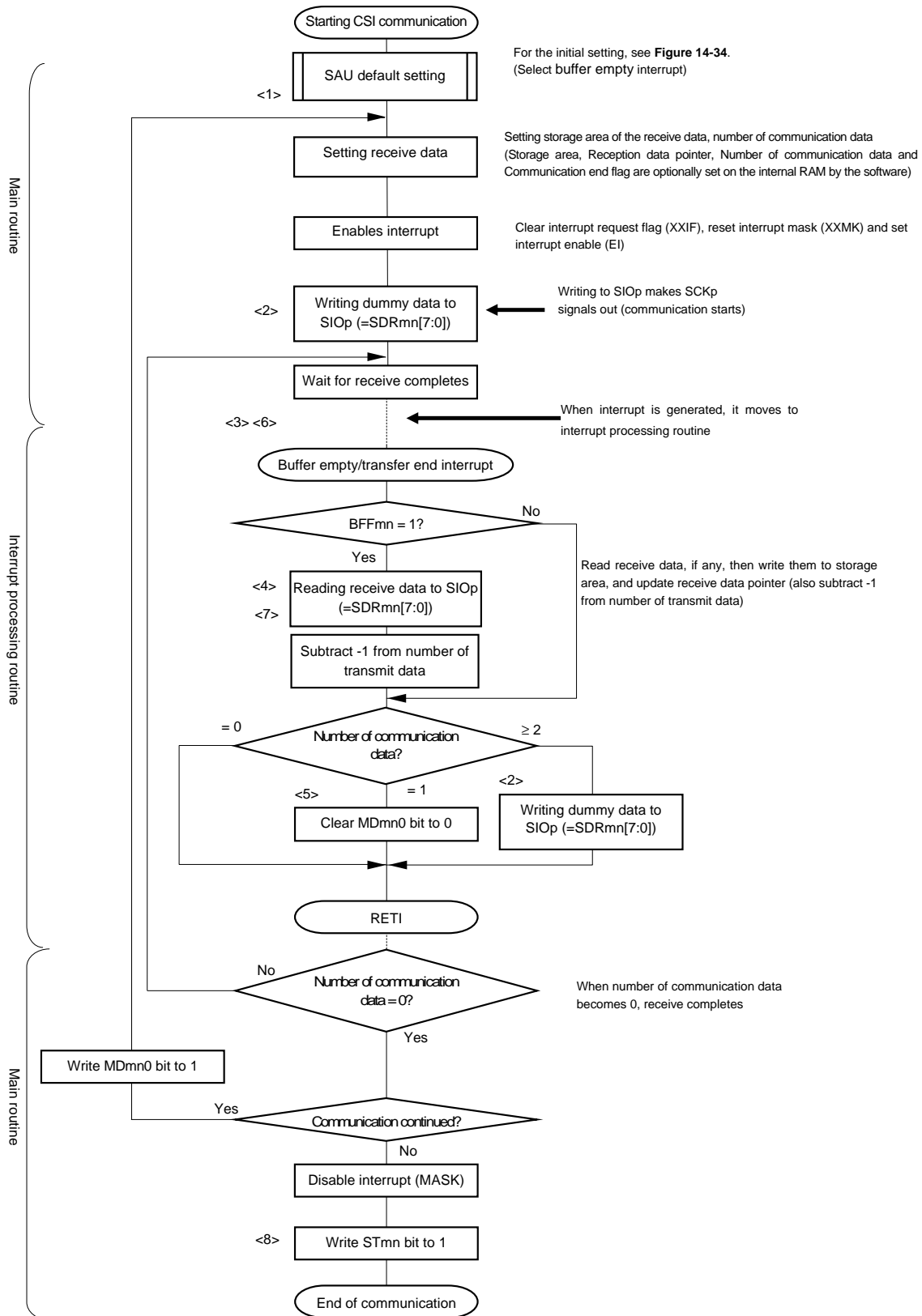


**Caution** The MDmn0 bit can be rewritten even during operation. However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 14-40 Flowchart of Master Reception (in Continuous Reception Mode)**.

**2.** m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02

Figure 14-40. Flowchart of Master Reception (in Continuous Reception Mode)



**Remark** <1> to <8> in the figure correspond to <1> to <8> in Figure 14-39 Timing Chart of Master Reception (in Continuous Reception Mode).

### 14.5.3 Master transmission/reception

Master transmission/reception is that the RL78 microcontroller outputs a transfer clock and transmits/receives data to/from other device.

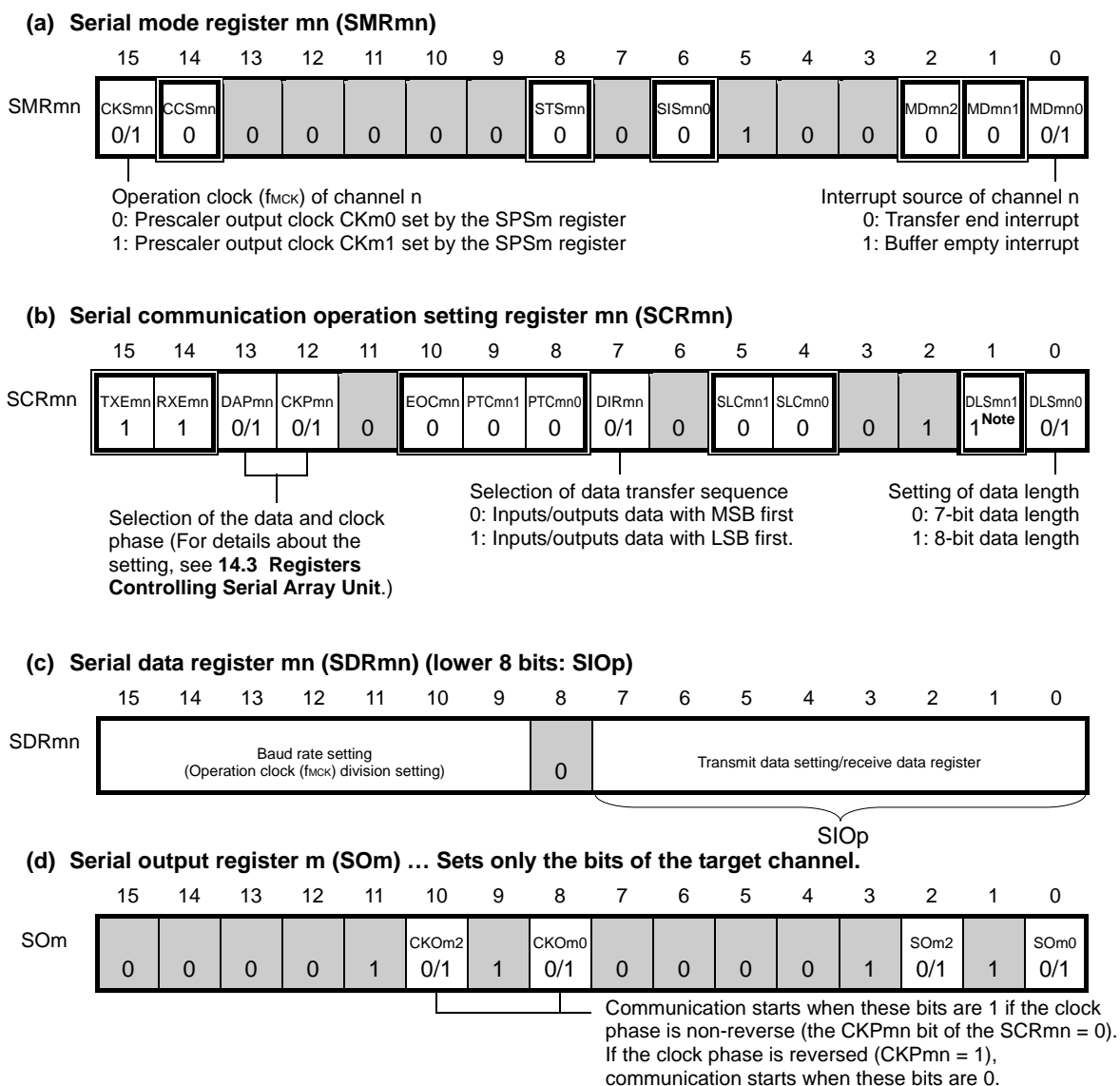
3-Wire Serial I/O	CSI00	CSI10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCK00, SI00, SO00	SCK10, SI10, SO10
Interrupt	INTCSI00	INTCSI10
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate <sup>Note</sup>	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 <ul style="list-style-type: none"> <li>• DAPmn = 0: Data I/O starts at the start of the operation of the serial clock.</li> <li>• DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.</li> </ul>	
Clock phase	Selectable by the CKPmn bit of the SCRmn register <ul style="list-style-type: none"> <li>• CKPmn = 0: Non-reverse</li> <li>• CKPmn = 1: Reverse</li> </ul>	
Data direction	MSB or LSB first	

**Note** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)**).

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02

(1) Register setting

Figure 14-41. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI10) (1/2)



**Note** Only provided for the SCR00 register. This bit is fixed to 1 for the other registers.

- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02
  - : Setting is fixed in the 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 14-41. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI10) (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	SOEm2 0/1	0	SOEm0 0/1

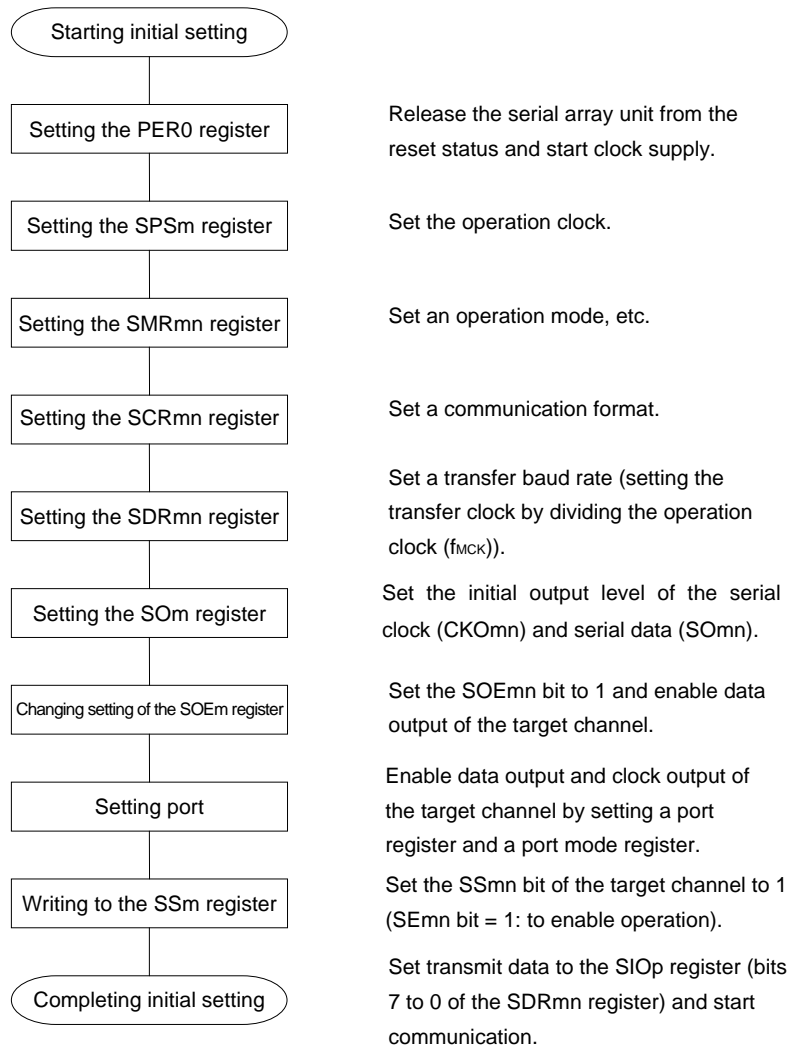
**(f) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.**

	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	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

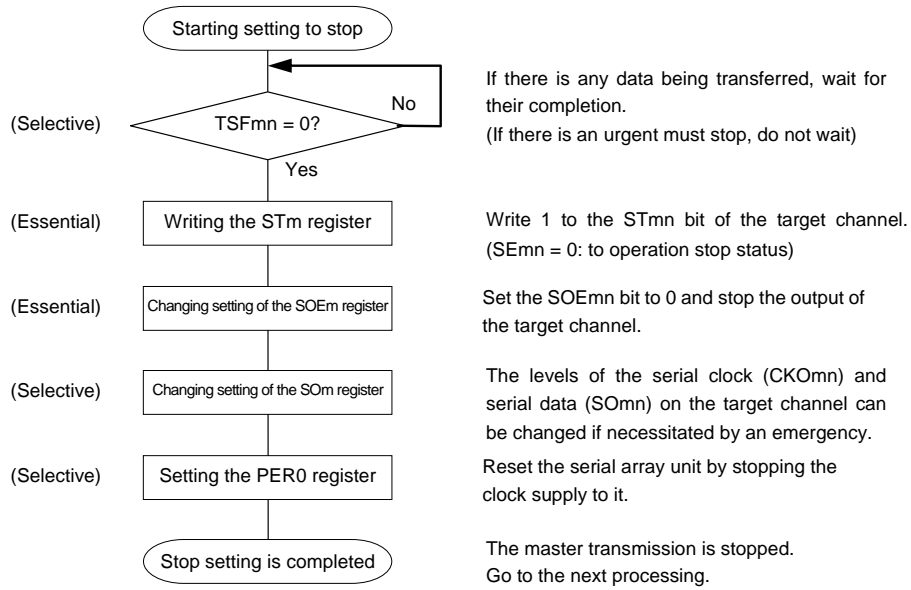
- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02
  - : Setting disabled (set to the initial value)
    - ×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
    - 0/1: Set to 0 or 1 depending on the usage of the user

## (2) Operation procedure

Figure 14-42. Initial Setting Procedure for Master Transmission/Reception

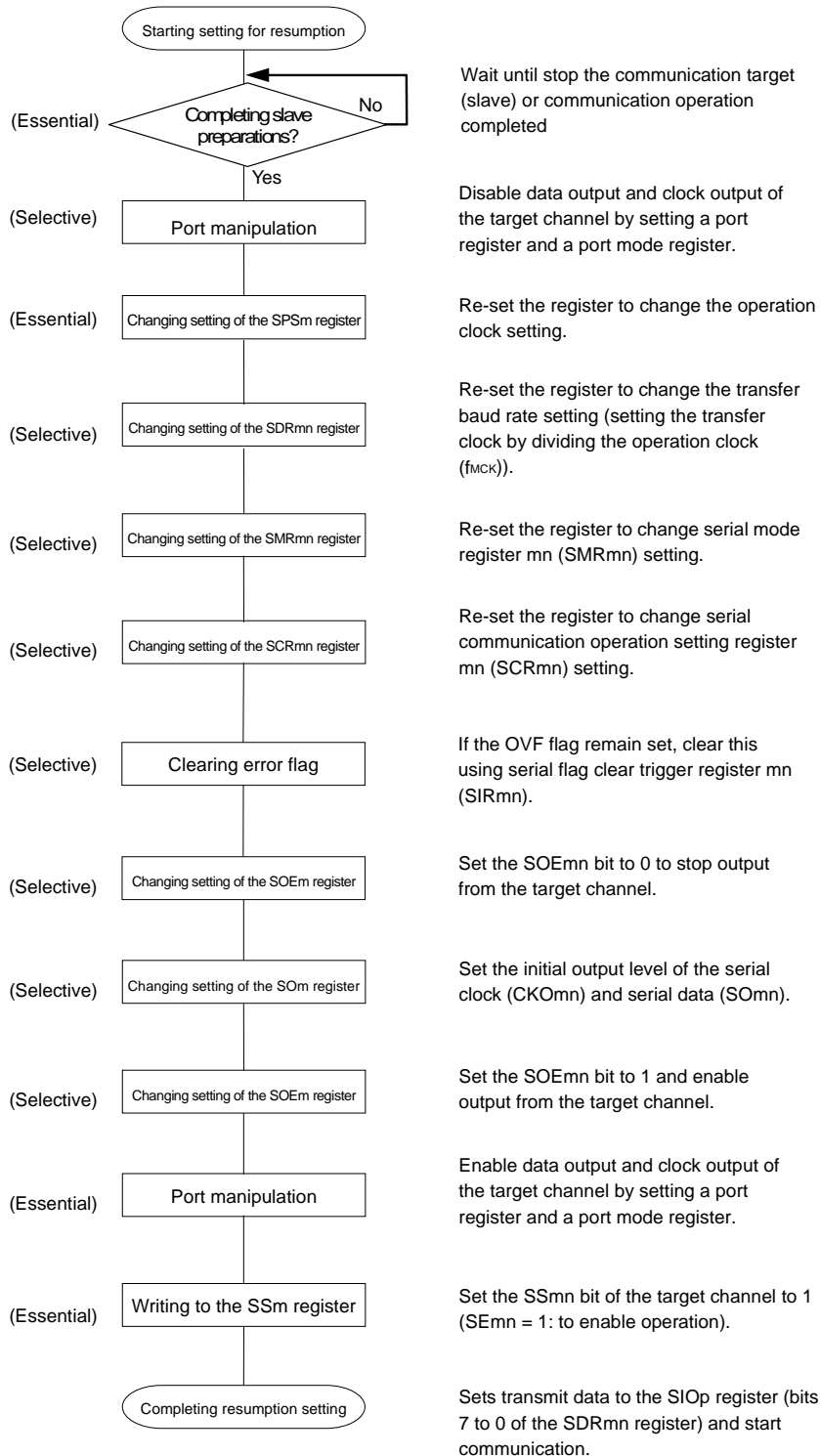


**Figure 14-43. Procedure for Stopping Master Transmission/Reception**



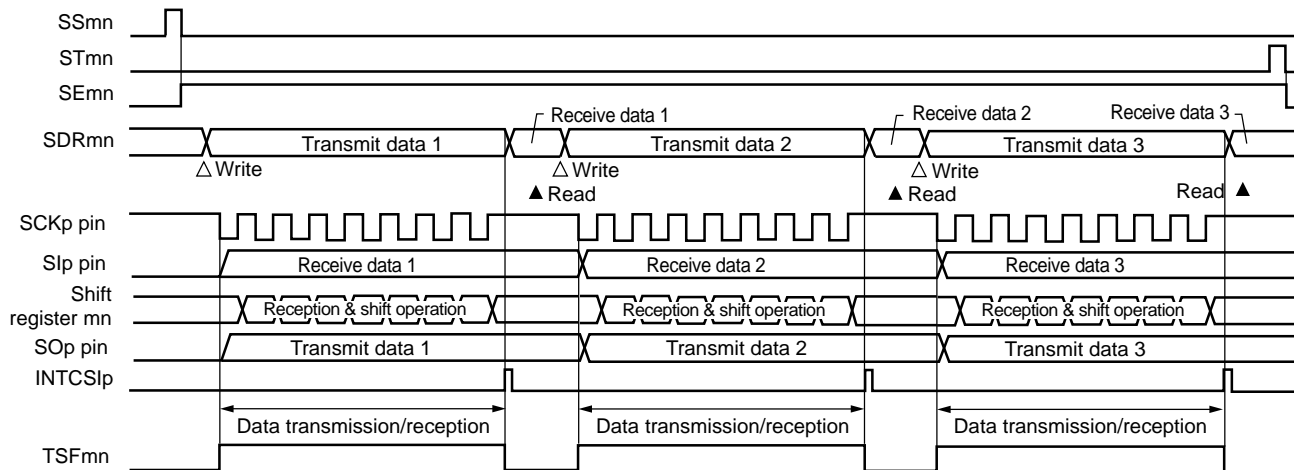


**Figure 14-44. Procedure for Resuming Master Transmission/Reception**



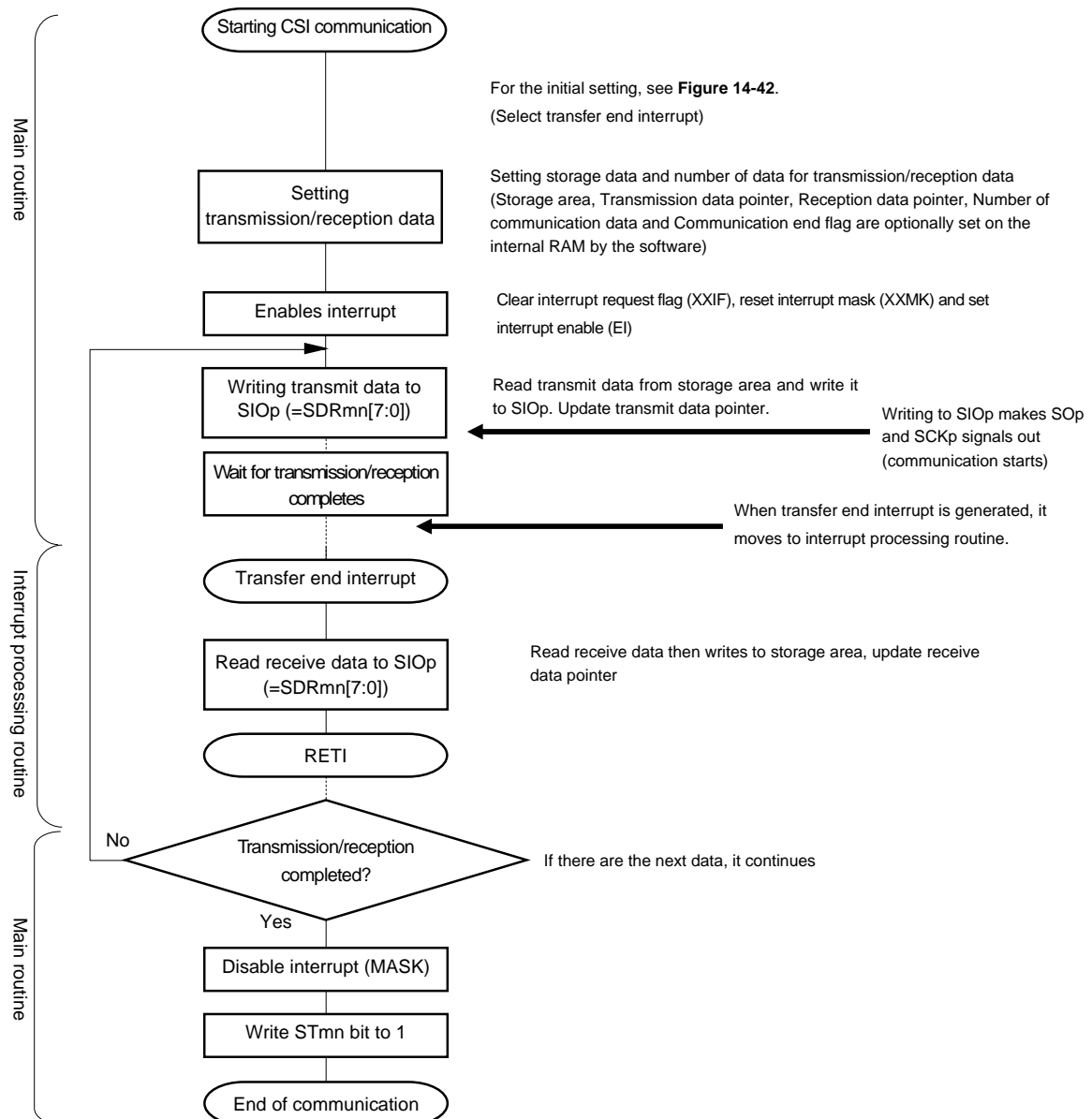
(3) Processing flow (in single-transmission/reception mode)

Figure 14-45. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)



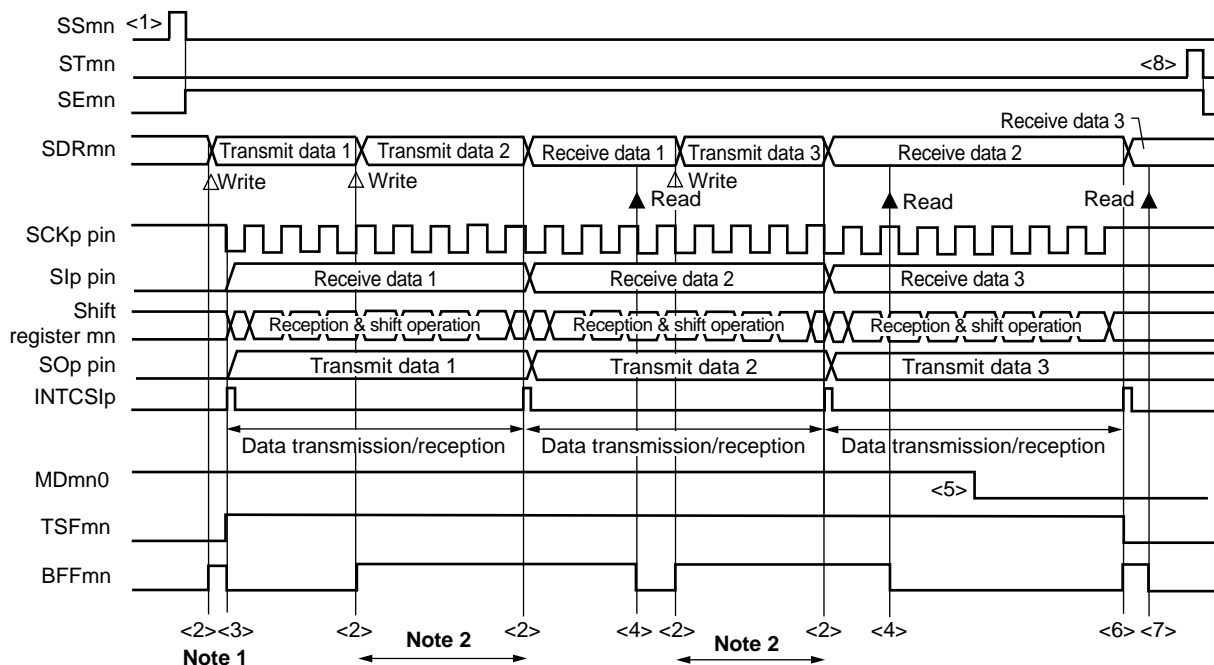
**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02

Figure 14-46. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)



(4) Processing flow (in continuous transmission/reception mode)

Figure 14-47. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)

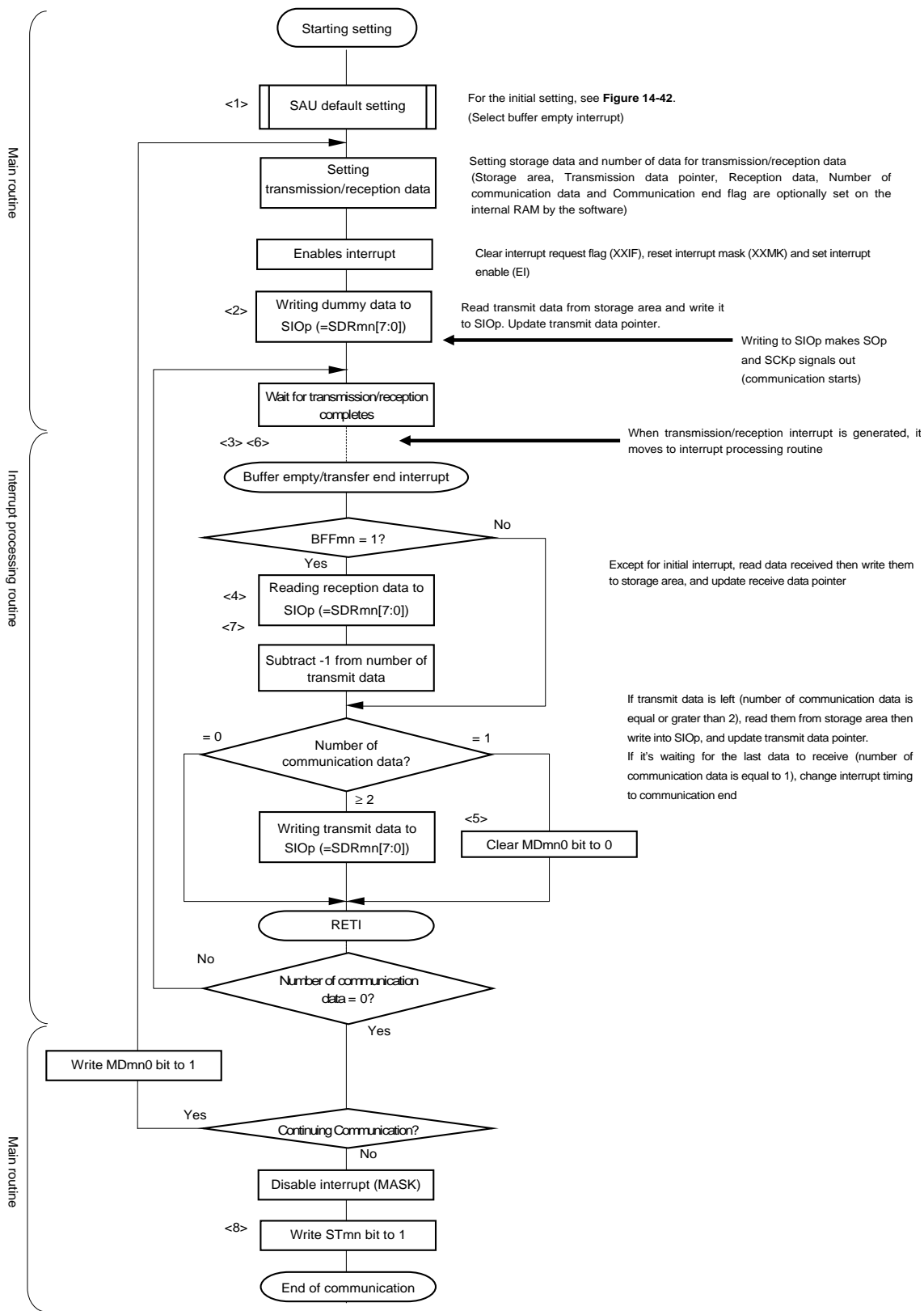


- Notes**
1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.
  2. The transmit data can be read by reading the SDRmn register during this period. 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 14-48 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.
  2. m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02

Figure 14-48. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



**Remark** <1> to <8> in the figure correspond to <1> to <8> in **Figure 14-47 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

#### 14.5.4 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.

3-Wire Serial I/O	CSI00	CSI10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCK00, SO00	SCK10, SO10
Interrupt	INTCSI00	INTCSI10
	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] <sup>Notes 1, 2</sup>	
Data phase	Selectable by the DAPmn bit of the SCRmn register <ul style="list-style-type: none"> <li>• DAPmn = 0: Data output starts from the start of the operation of the serial clock.</li> <li>• DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.</li> </ul>	
Clock phase	Selectable by the CKPmn bit of the SCRmn register <ul style="list-style-type: none"> <li>• CKPmn = 0: Non-reverse</li> <li>• CKPmn = 1: Reverse</li> </ul>	
Data direction	MSB or LSB first	

**Notes 1.** Because the external serial clock input to the SCK00, SCK10 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 peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)**).

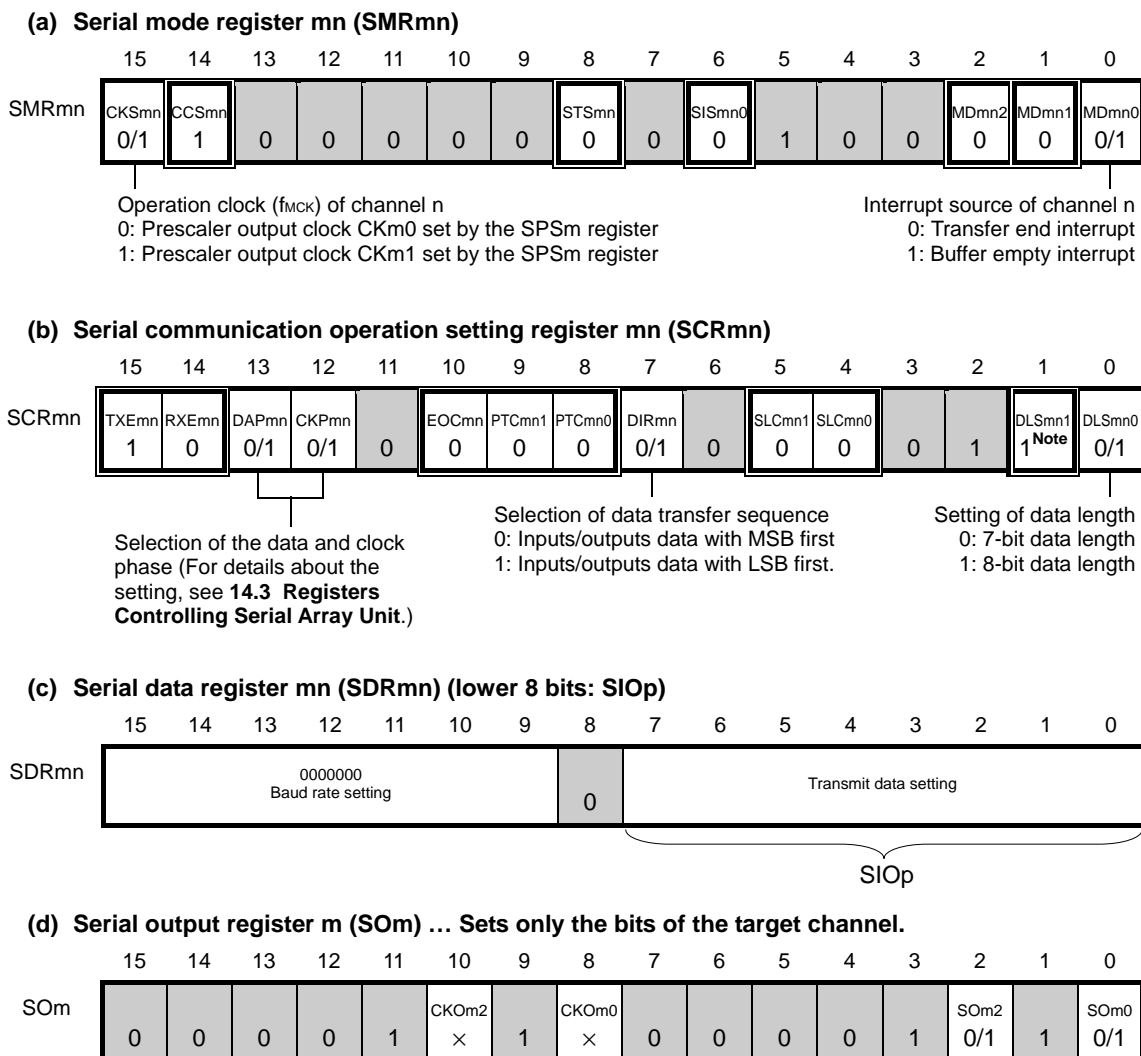
**Remarks 1.**  $f_{MCK}$ : Operation clock frequency of target channel

$f_{SCK}$ : Serial clock frequency

**2.** m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02

(1) Register setting

Figure 14-49. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI10) (1/2)



**Note** Only provided for the SCR00 register. This bit is fixed to 1 for the other registers.

- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02
  - : Setting is fixed in the 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 14-49. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI10) (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	SOEm2 0/1	0	SOEm0 0/1

**(f) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.**

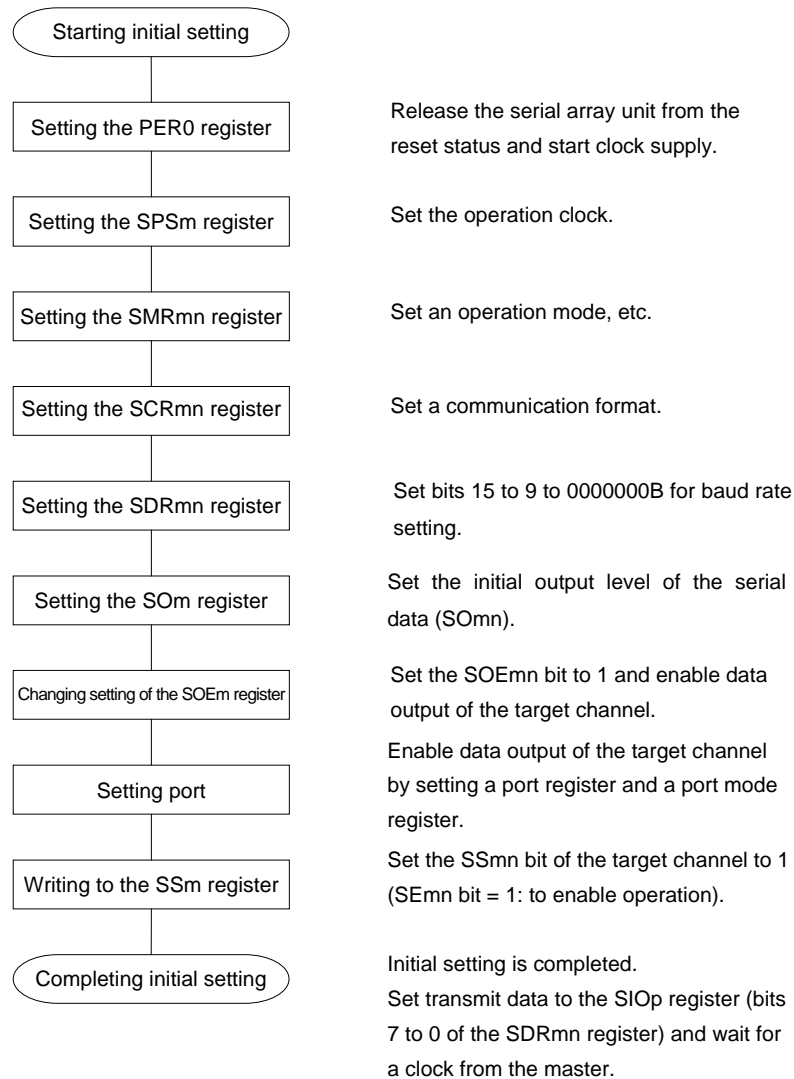
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

- Remarks**
1. m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02
  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 14-50. Initial Setting Procedure for Slave Transmission



**Figure 14-51. Procedure for Stopping Slave Transmission**

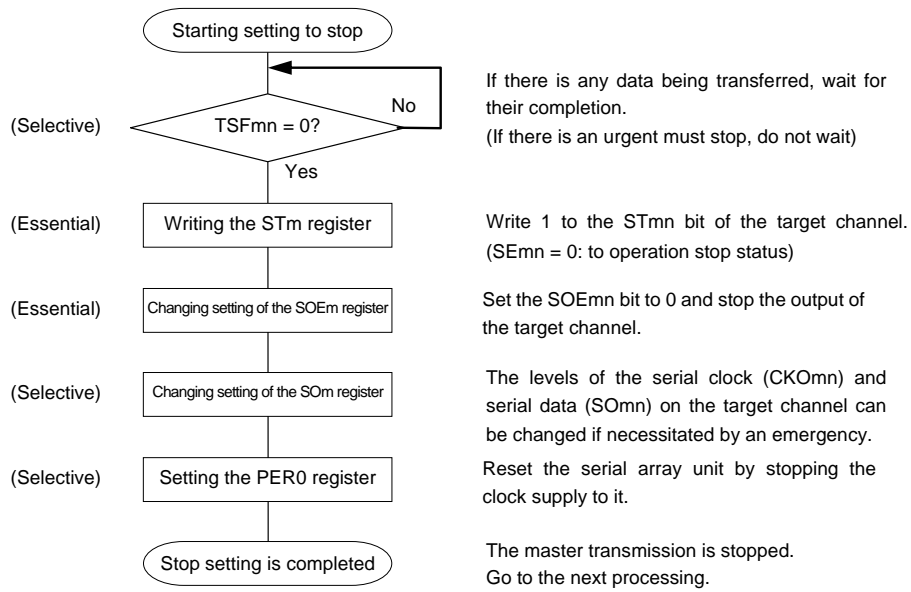
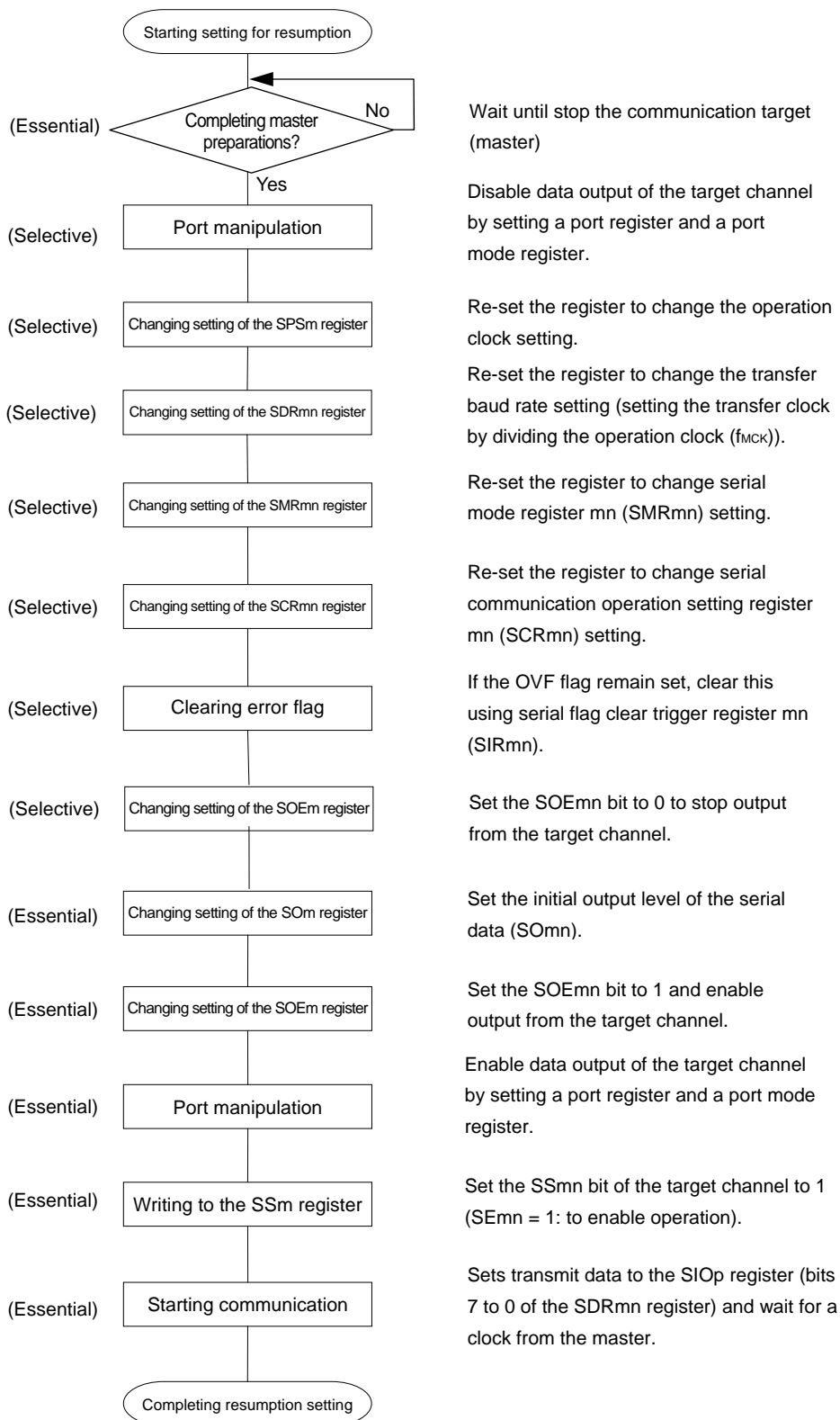


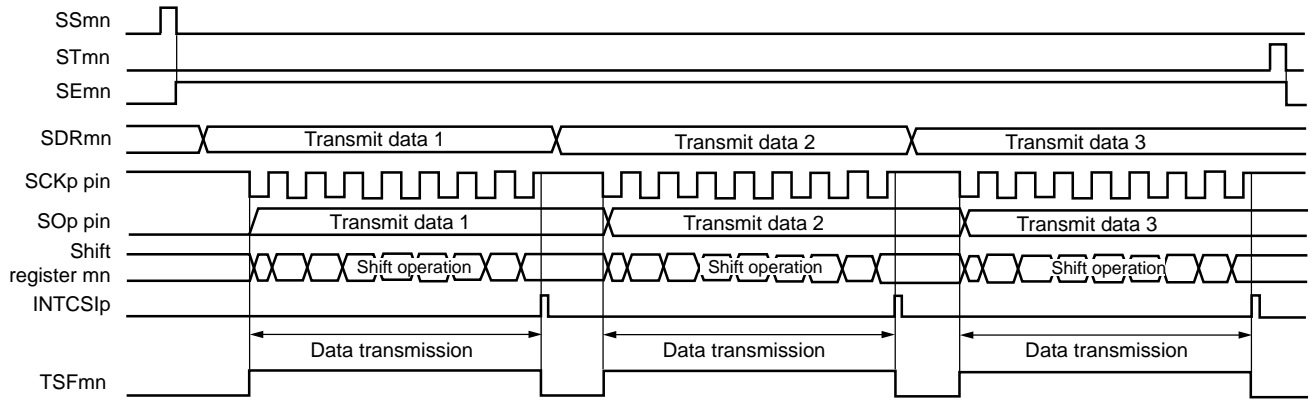
Figure 14-52. Procedure for Resuming Slave Transmission



**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.

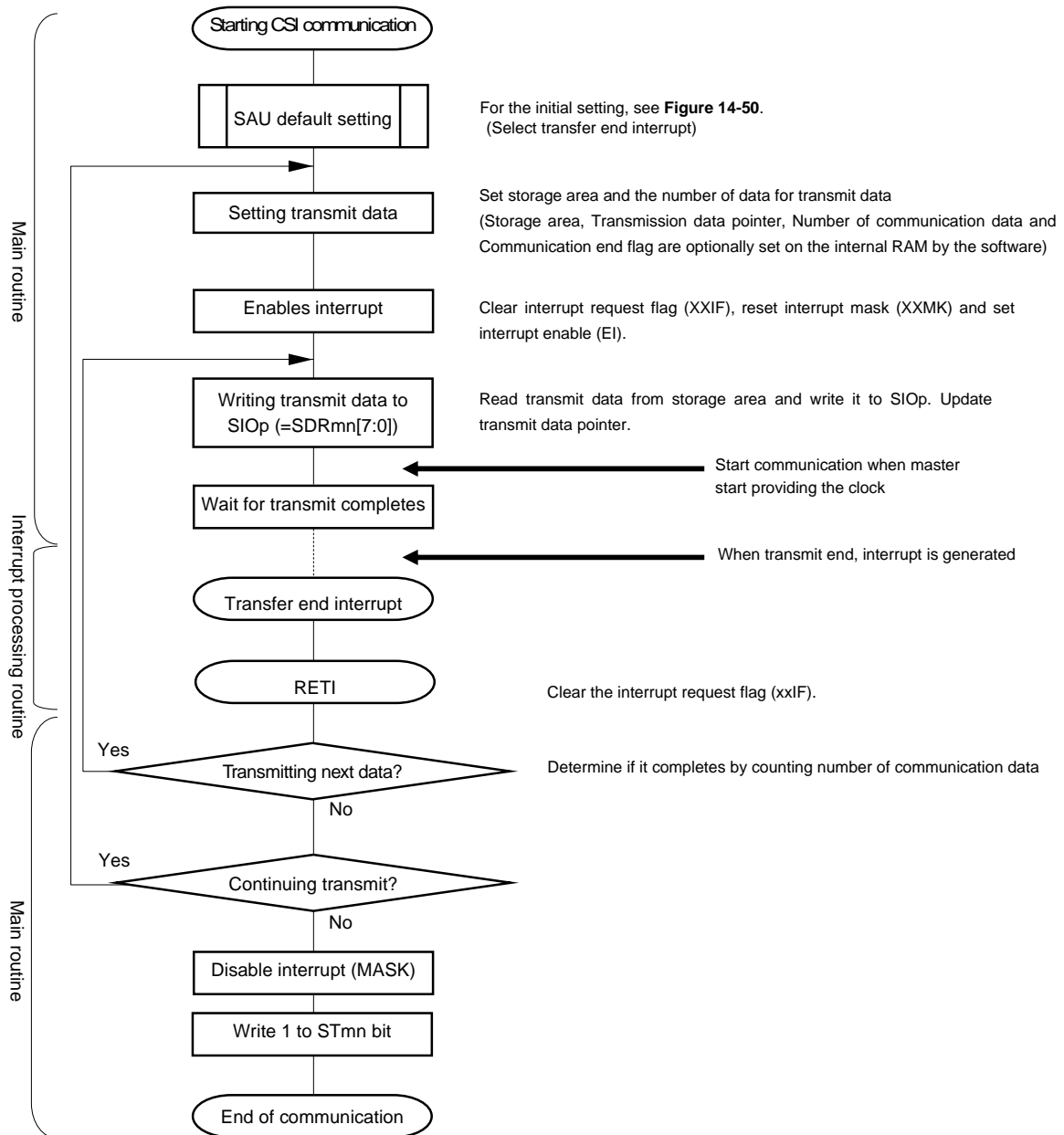
(3) Processing flow (in single-transmission mode)

Figure 14-53. Timing Chart of Slave Transmission (in Single-Transmission Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)



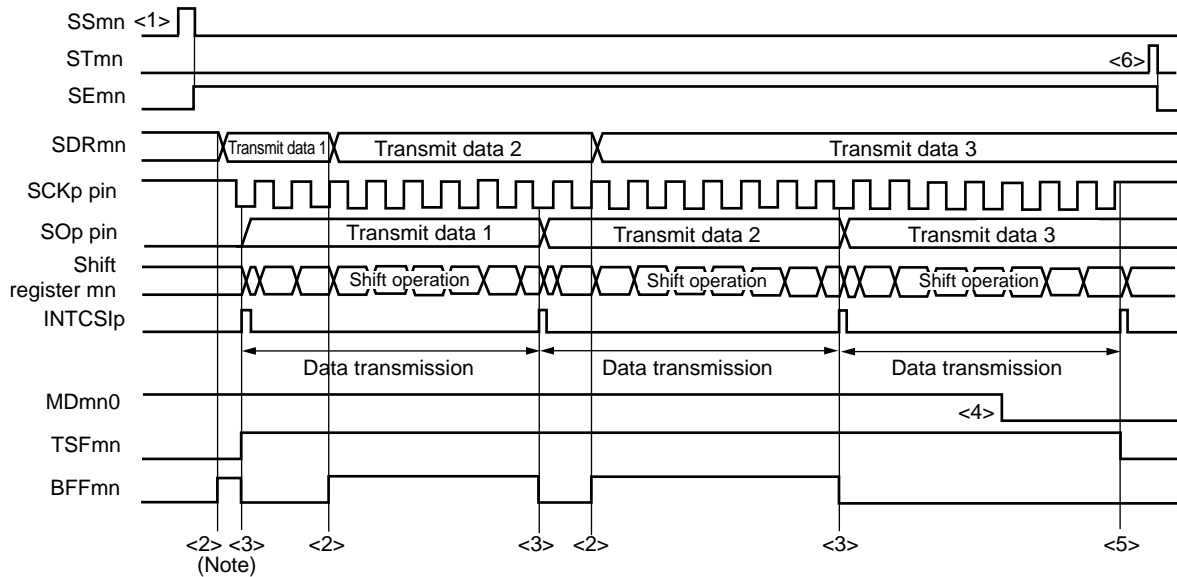
**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02

Figure 14-54. Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 14-55. Timing Chart of Slave Transmission (in Continuous Transmission Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)

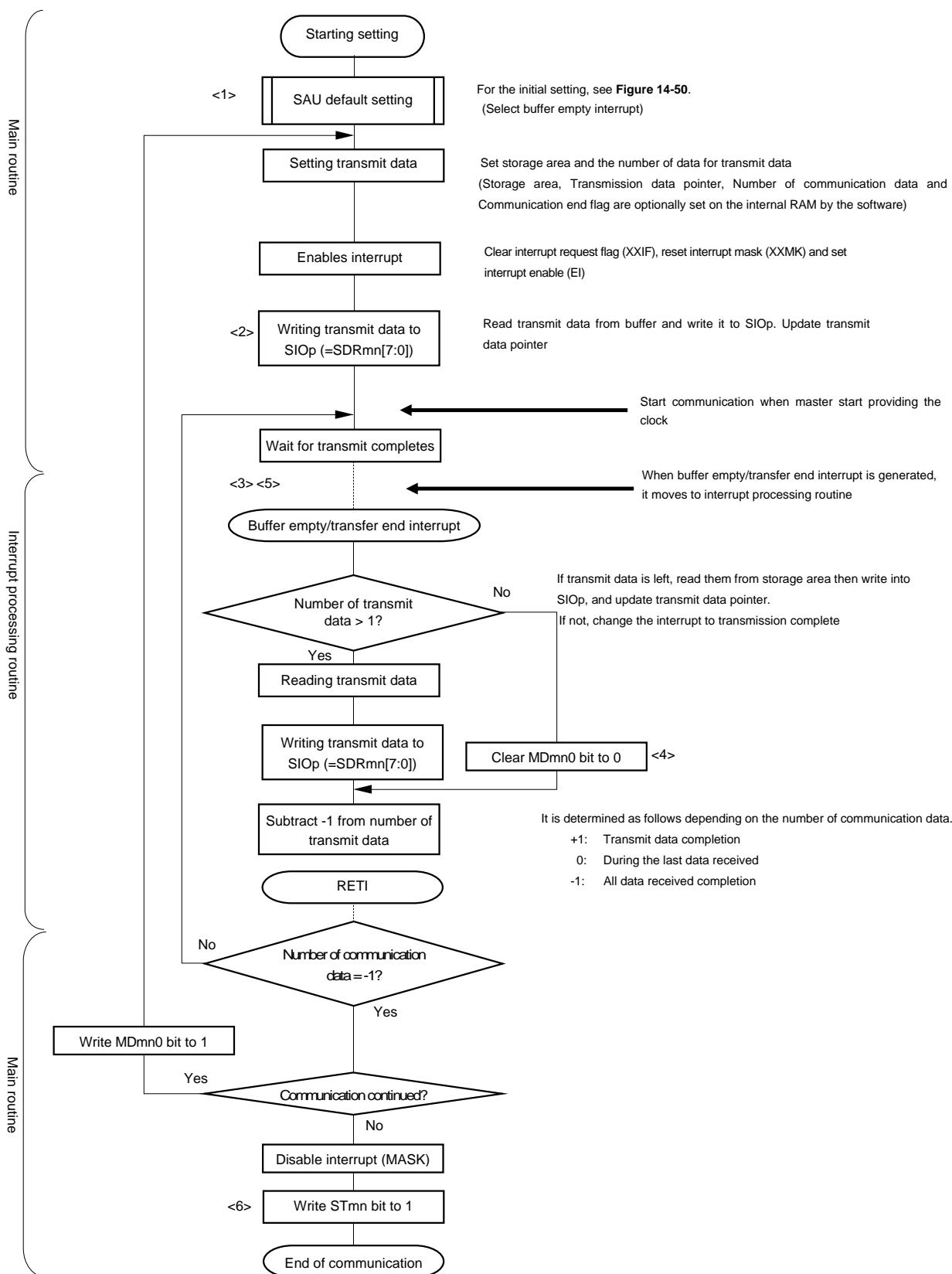


**Note** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

**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, 2), p: CSI number (p = 00, 10), mn = 00, 02

Figure 14-56. Flowchart of Slave Transmission (in Continuous Transmission Mode)



**Remark** <1> to <6> in the figure correspond to <1> to <6> in Figure 14-55 Timing Chart of Slave Transmission (in Continuous Transmission Mode).

### 14.5.5 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.

3-Wire Serial I/O	CSI00	CSI10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCK00, SI00	SCK10, SI10
Interrupt	INTCSI00	INTCSI10
	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] <sup>Notes 1, 2</sup>	
Data phase	Selectable by the DAPmn bit of the SCRmn register <ul style="list-style-type: none"> <li>• DAPmn = 0: Data input starts from the start of the operation of the serial clock.</li> <li>• DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.</li> </ul>	
Clock phase	Selectable by the CKPmn bit of the SCRmn register <ul style="list-style-type: none"> <li>• CKPmn = 0: Non-reverse</li> <li>• CKPmn = 1: Reverse</li> </ul>	
Data direction	MSB or LSB first	

**Notes 1.** Because the external serial clock input to the SCK00, SCK10 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 peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)**).

**Remarks 1.**  $f_{MCK}$ : Operation clock frequency of target channel

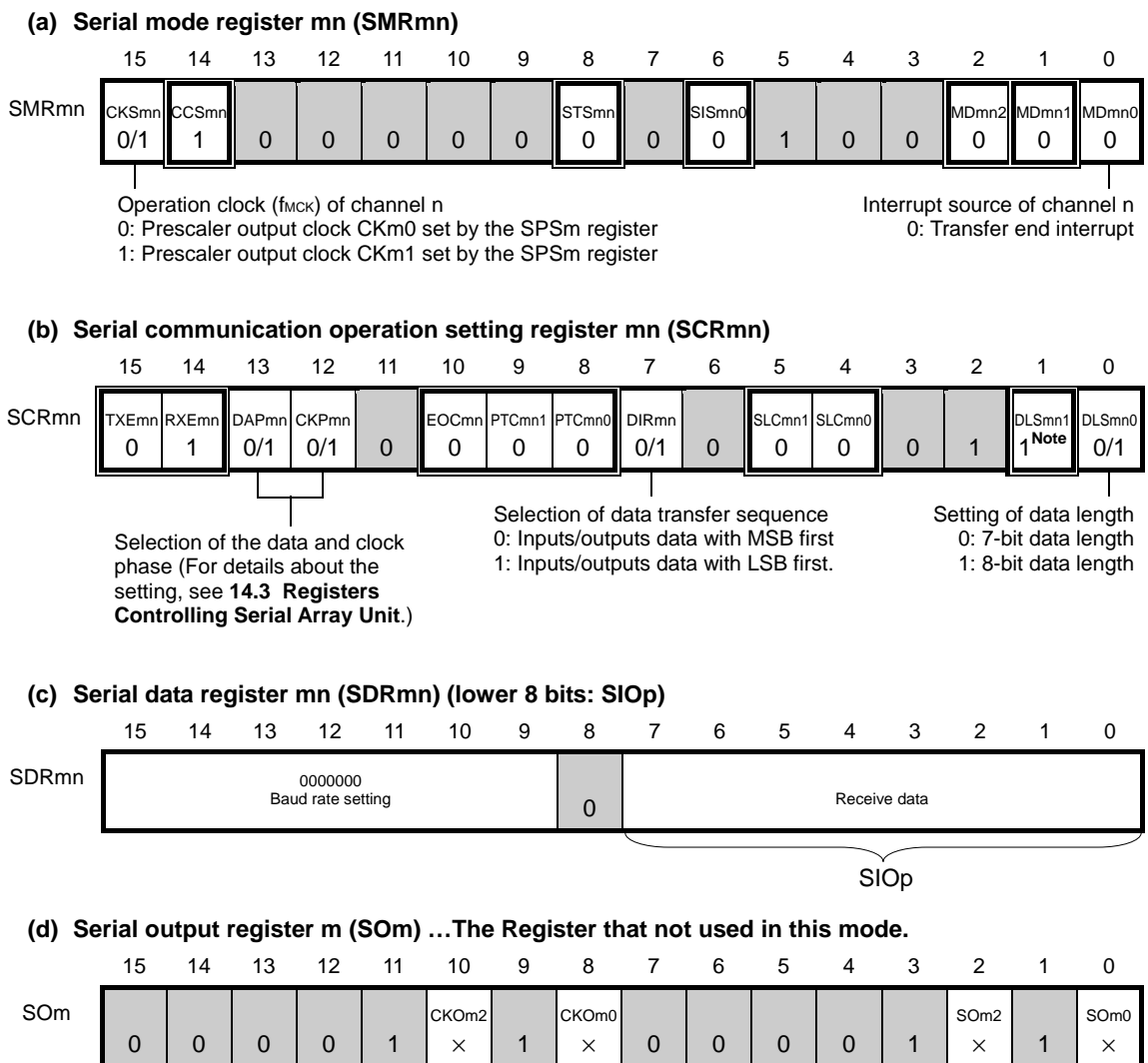
$f_{SCK}$ : Serial clock frequency

**2.** m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02



(1) Register setting

Figure 14-57. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI10) (1/2)



**Note** Only provided for the SCR00 register. This bit is fixed to 1 for the other registers.

- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02
  - : Setting is fixed in the 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 14-57. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI10) (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	SOEm2 ×	0	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	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02
  - : 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 14-58. Initial Setting Procedure for Slave Reception

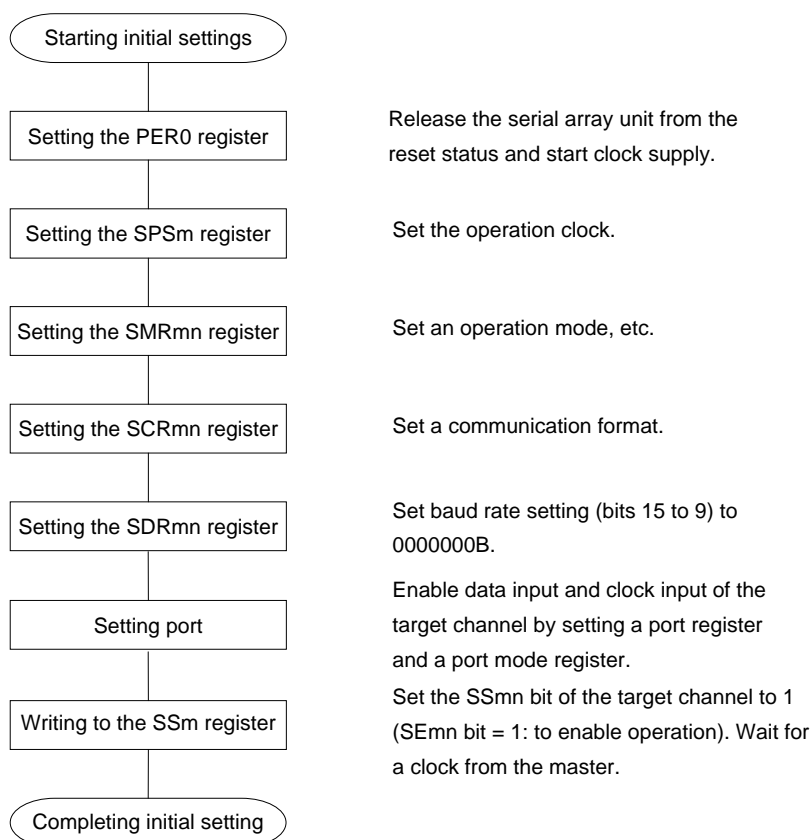
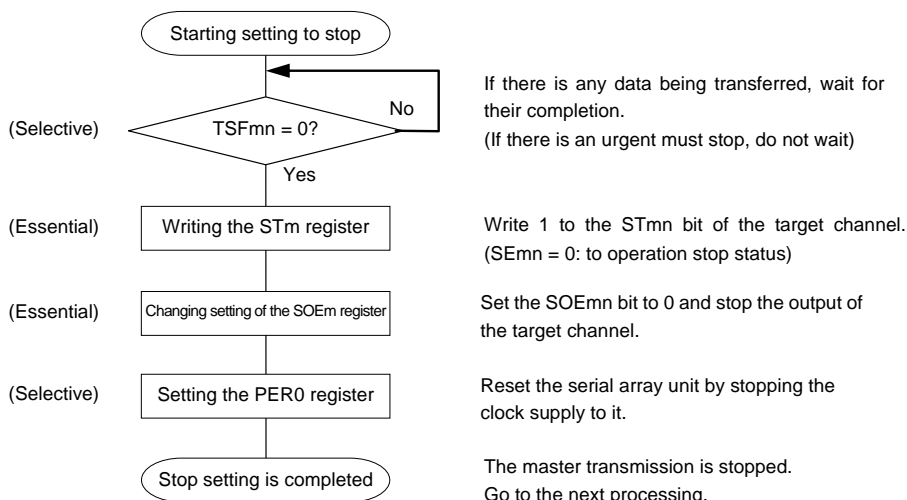
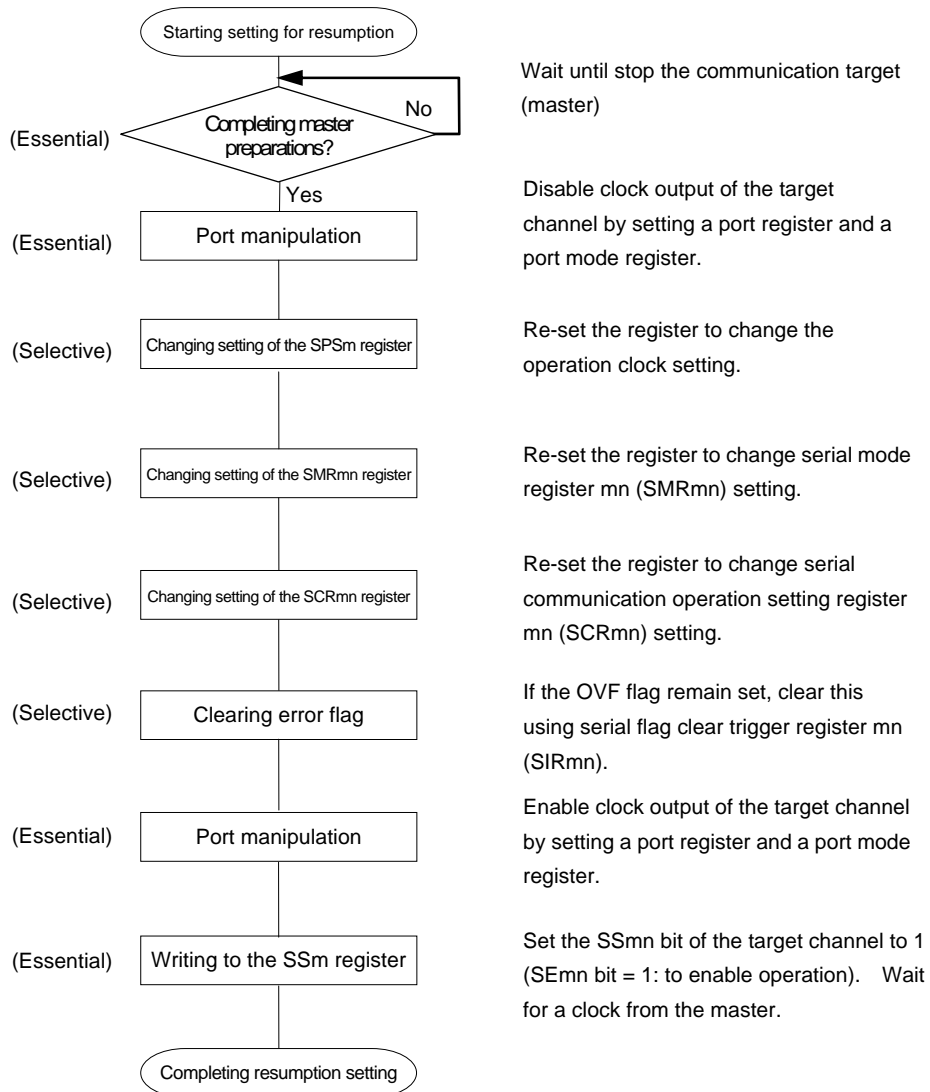


Figure 14-59. Procedure for Stopping Slave Reception



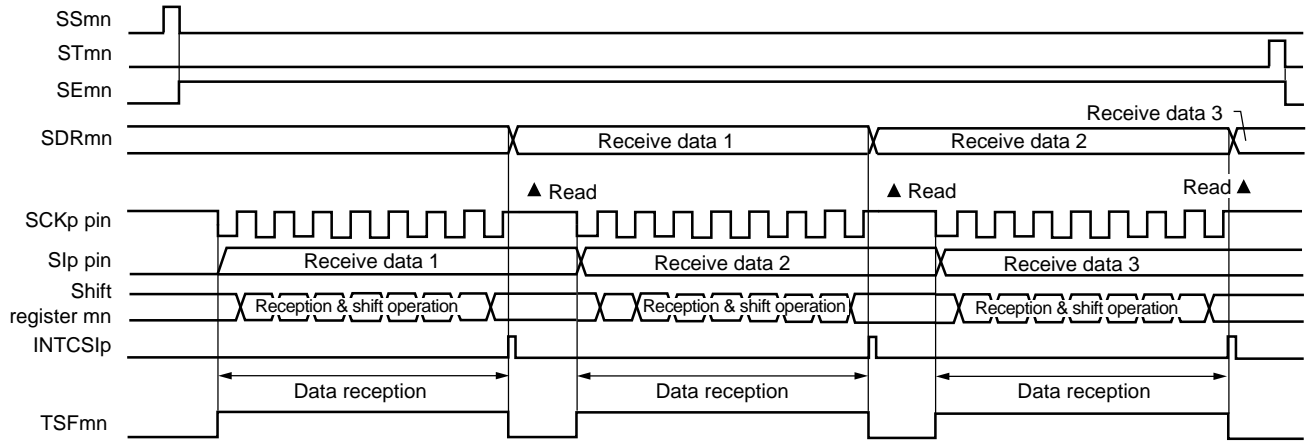
**Figure 14-60. Procedure for Resuming Slave Reception**



**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.

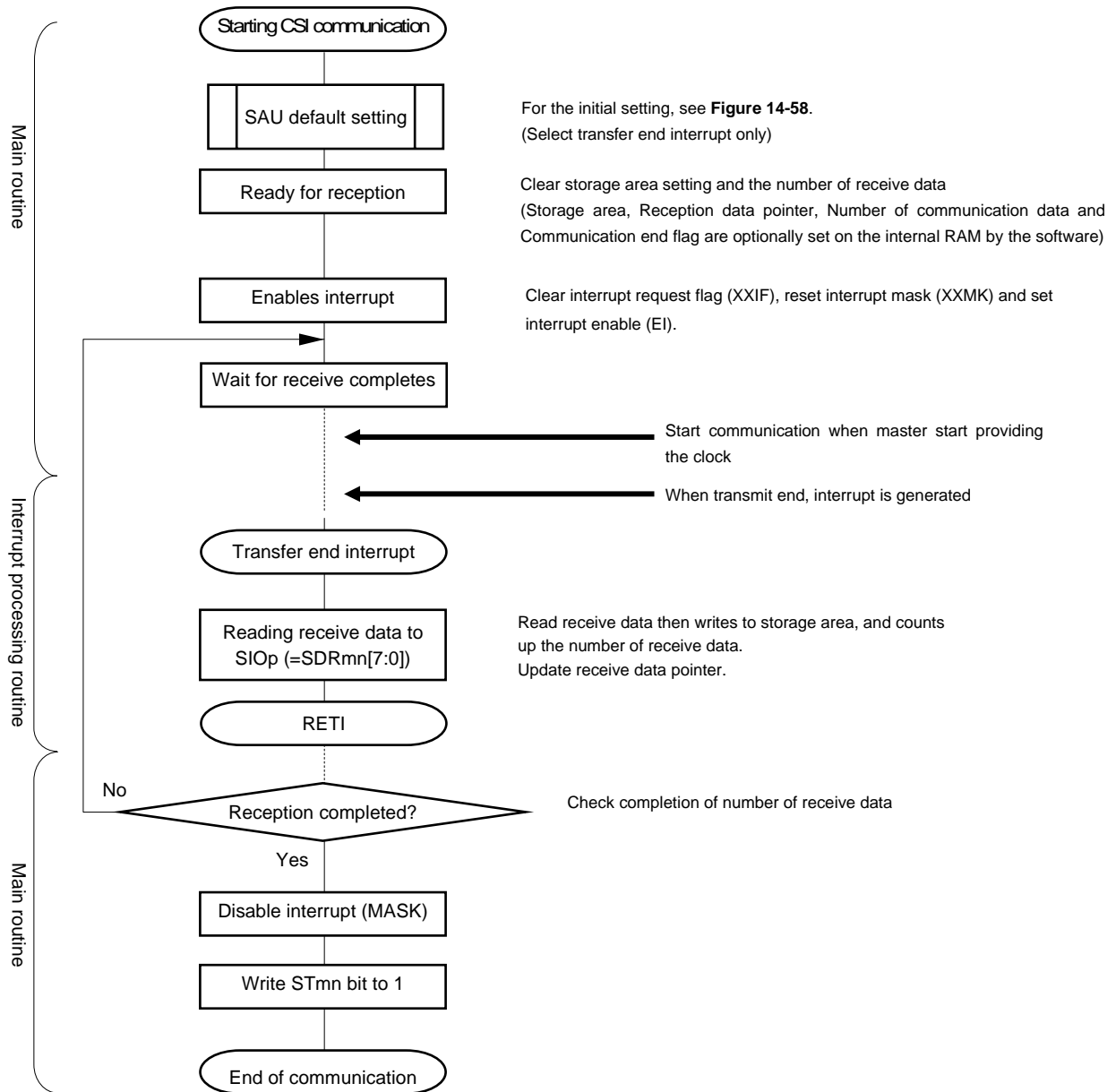
(3) Processing flow (in single-reception mode)

Figure 14-61. 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, 2), p: CSI number (p = 00, 10), mn = 00, 02

**Figure 14-62. Flowchart of Slave Reception (in Single-Reception Mode)**



### 14.5.6 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.

3-Wire Serial I/O	CSI00	CSI10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCK00, SI00, SO00	SCK10, SI10, SO10
Interrupt	INTCSI00	INTCSI10
	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] <sup>Notes 1, 2</sup>	
Data phase	Selectable by the DAPmn bit of the SCRmn register <ul style="list-style-type: none"> <li>• DAPmn = 0: Data I/O starts from the start of the operation of the serial clock.</li> <li>• DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.</li> </ul>	
Clock phase	Selectable by the CKPmn bit of the SCRmn register <ul style="list-style-type: none"> <li>• CKPmn = 0: Non-reverse</li> <li>• CKPmn = 1: Reverse</li> </ul>	
Data direction	MSB or LSB first	

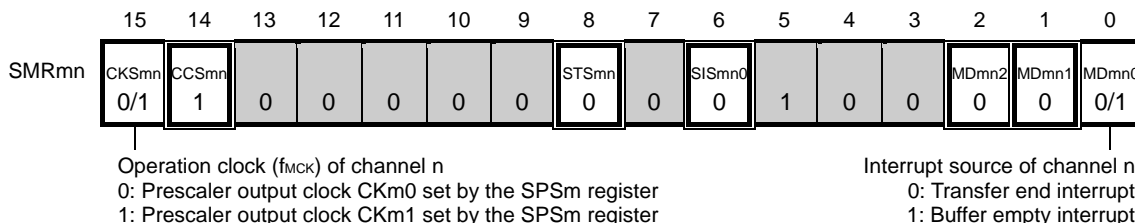
- Notes 1.** Because the external serial clock input to the SCK00, SCK10 pins is sampled internally and used, the fastest transfer rate is  $f_{MCK}/6$  [Hz].
- 2.** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)**).

- Remarks 1.**  $f_{MCK}$ : Operation clock frequency of target channel  
 $f_{CLK}$ : Serial clock frequency
- 2.** m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02

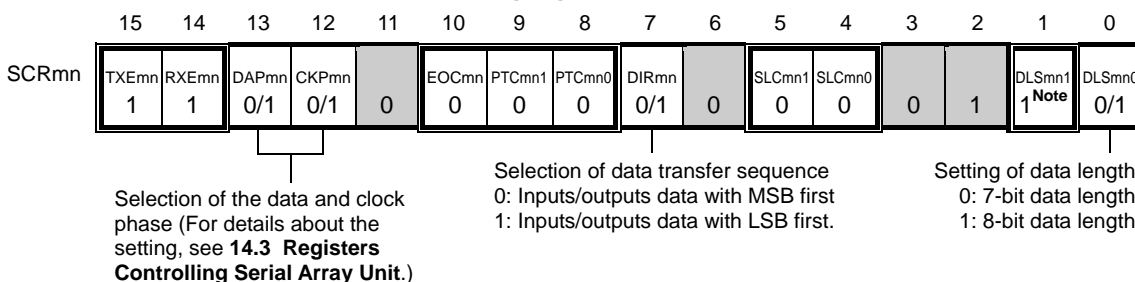
(1) Register setting

Figure 14-63. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI10) (1/2)

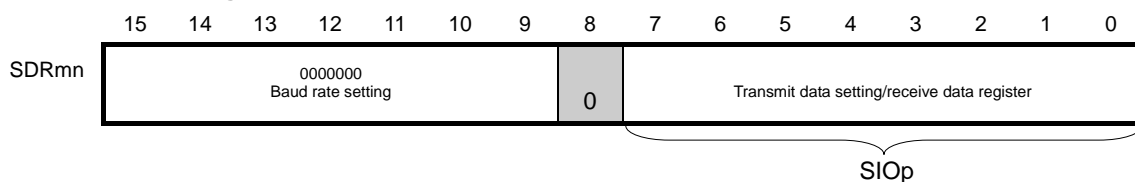
(a) Serial mode register mn (SMRmn)



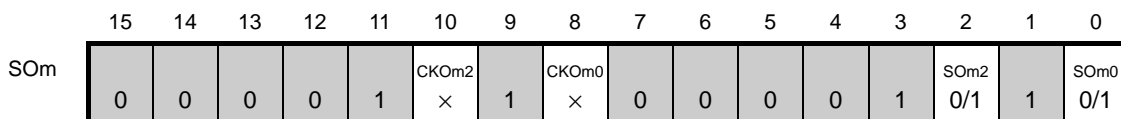
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOM) ... Sets only the bits of the target channel.



**Note** Only provided for the SCR00 register. This bit is fixed to 1 for the other registers.

**Caution** Be sure to set transmit data to the SIOp register before the clock from the master is started.

- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02
  - |     |   |
|-----|---|
|     | : Setting is fixed in the 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 14-63. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI10) (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	SOEm2 0/1	0	SOEm0 0/1

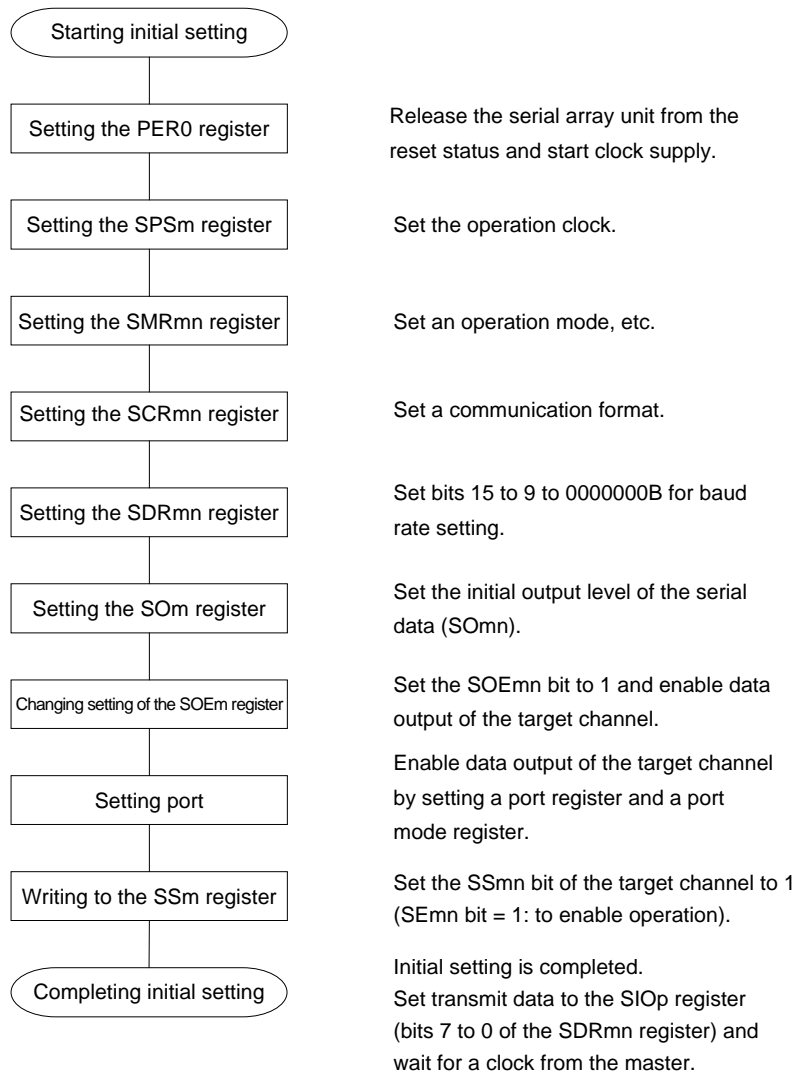
**(f) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.**

	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	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02
  - : 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 14-64. Initial Setting Procedure for Slave Transmission/Reception



**Figure 14-65. Procedure for Stopping Slave Transmission/Reception**

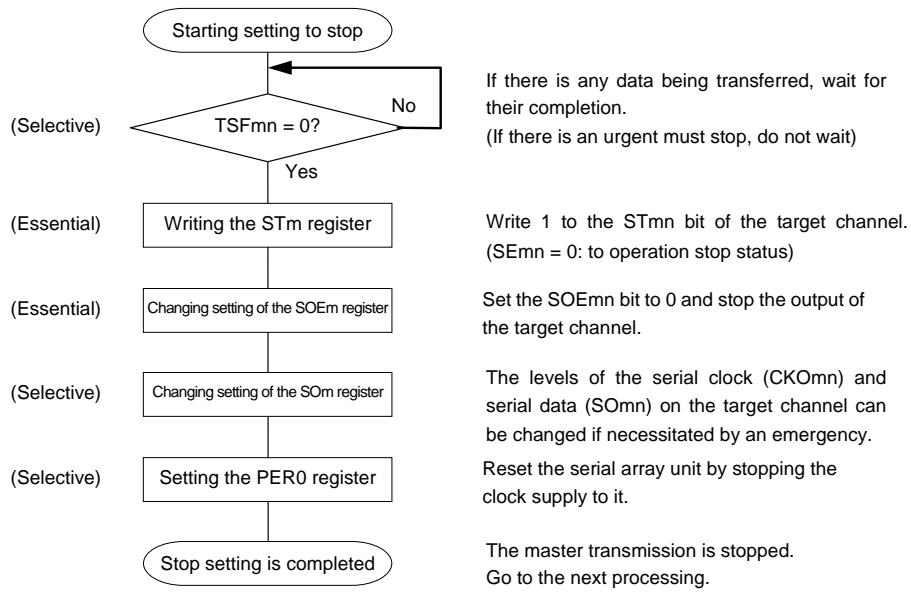
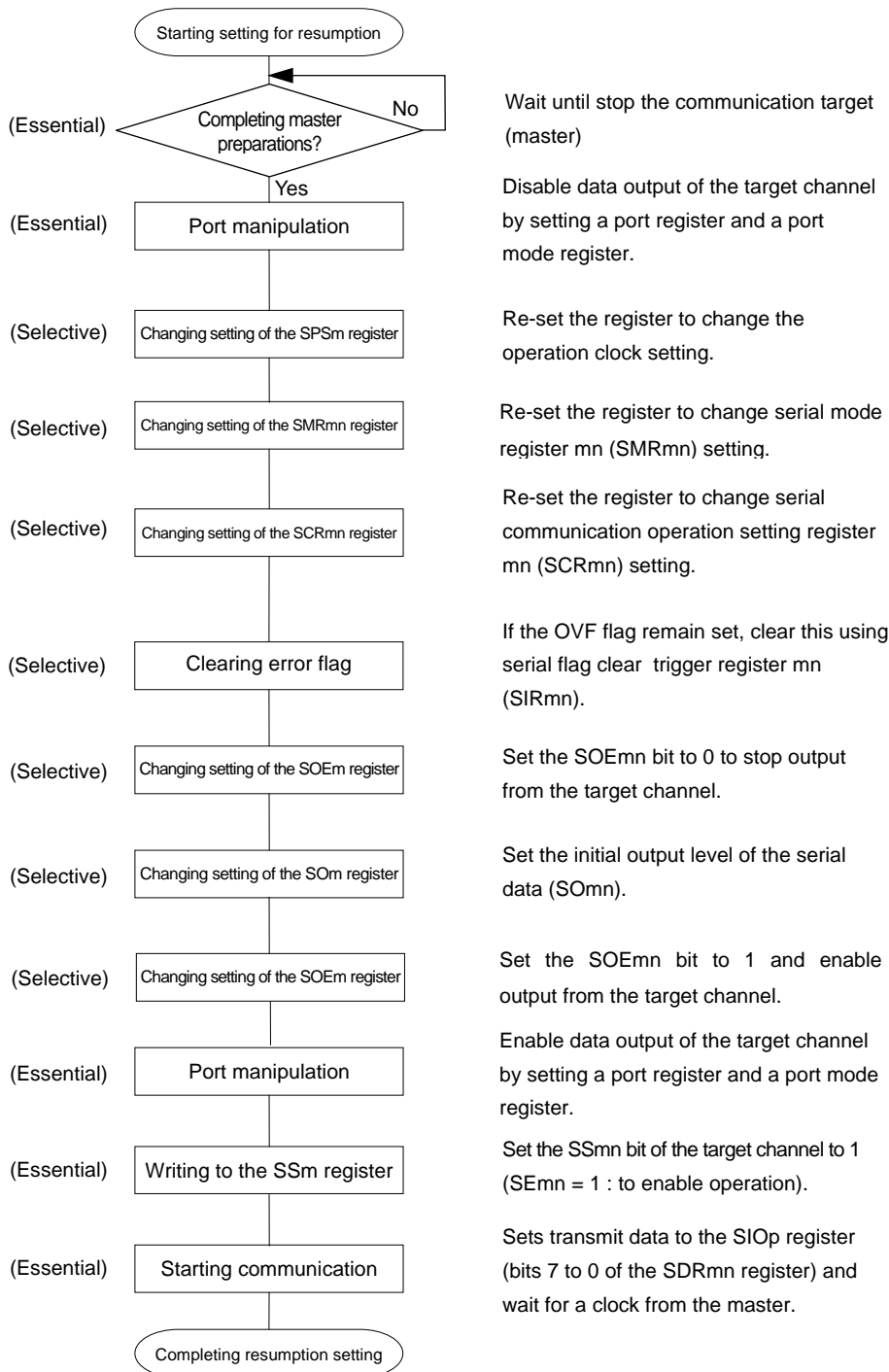


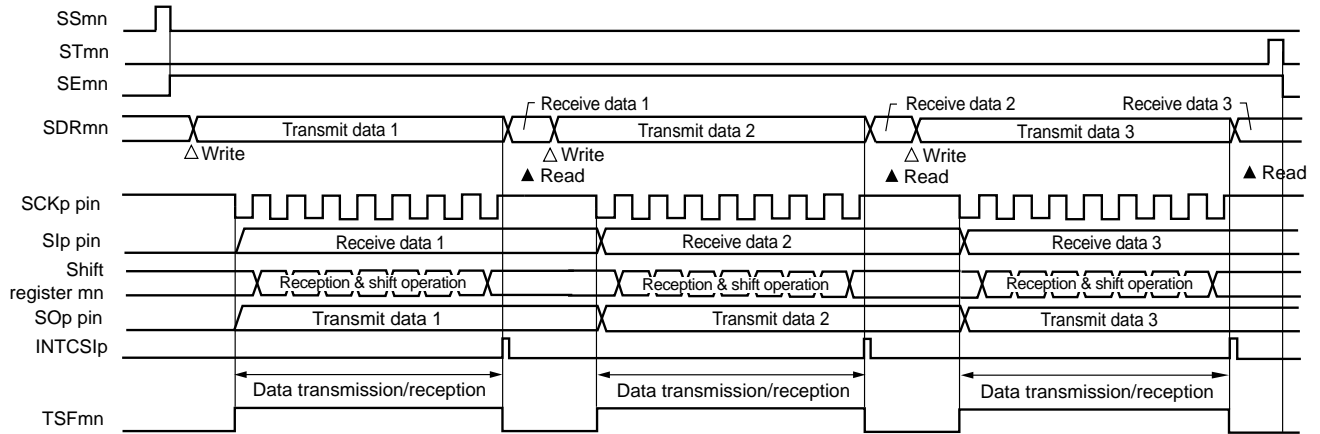
Figure 14-66. Procedure for Resuming Slave Transmission/Reception



- 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.

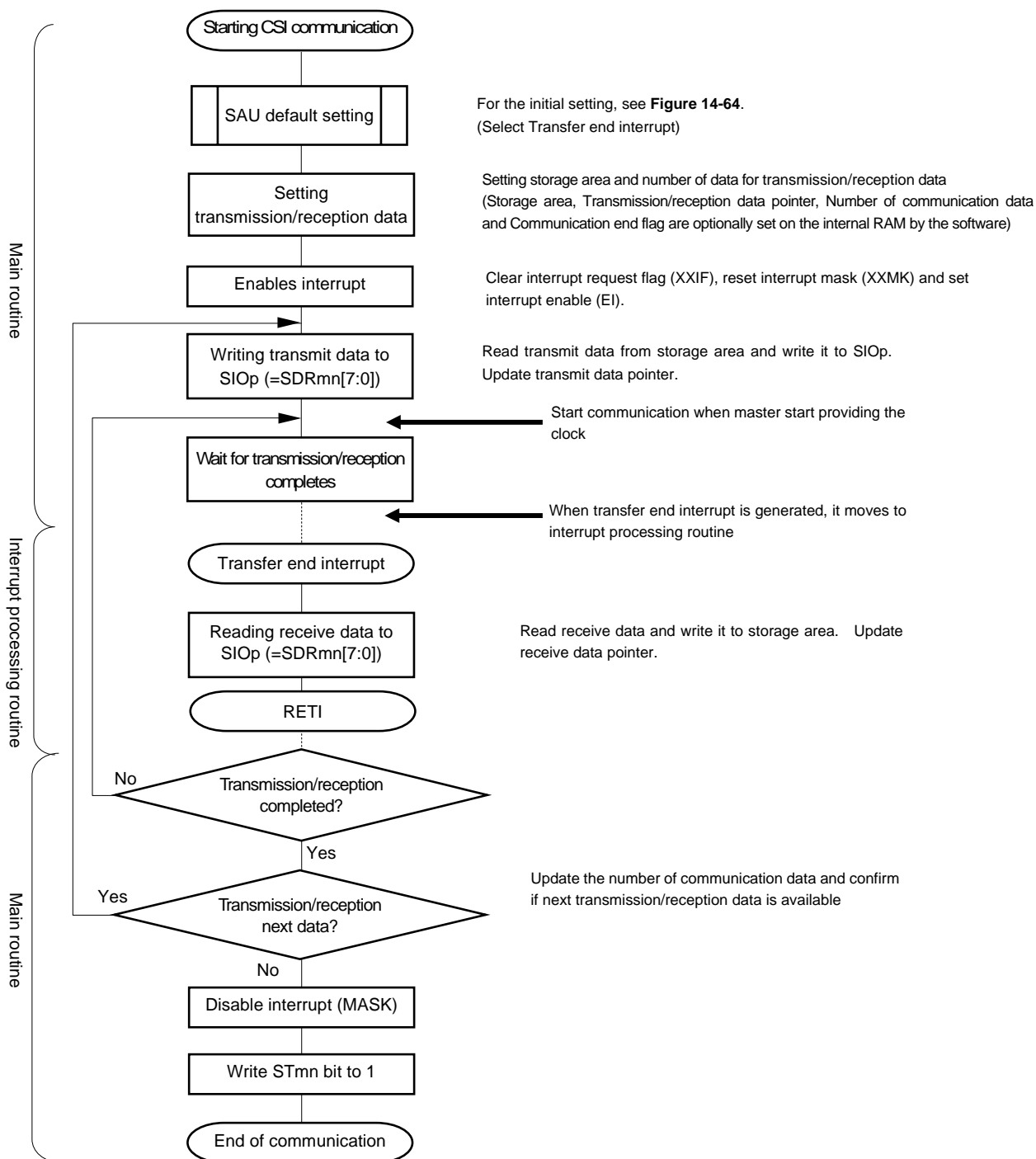
(3) Processing flow (in single-transmission/reception mode)

Figure 14-67. 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, 2), p: CSI number (p = 00, 10), mn = 00, 02

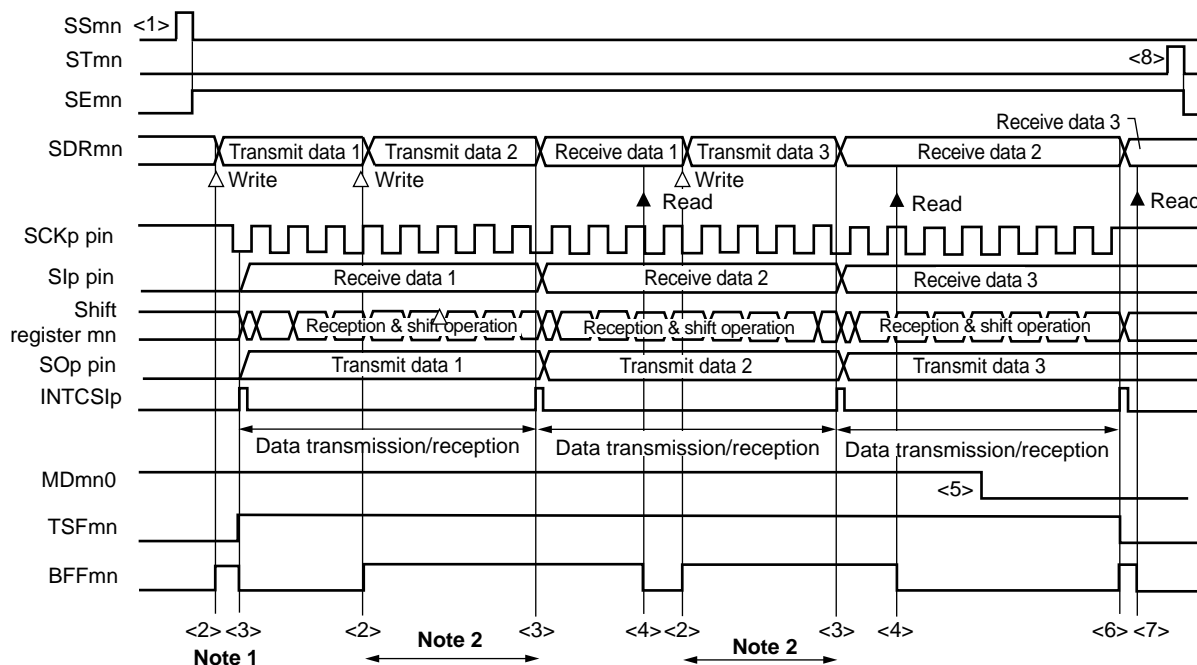
Figure 14-68. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)



**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 14-69. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)

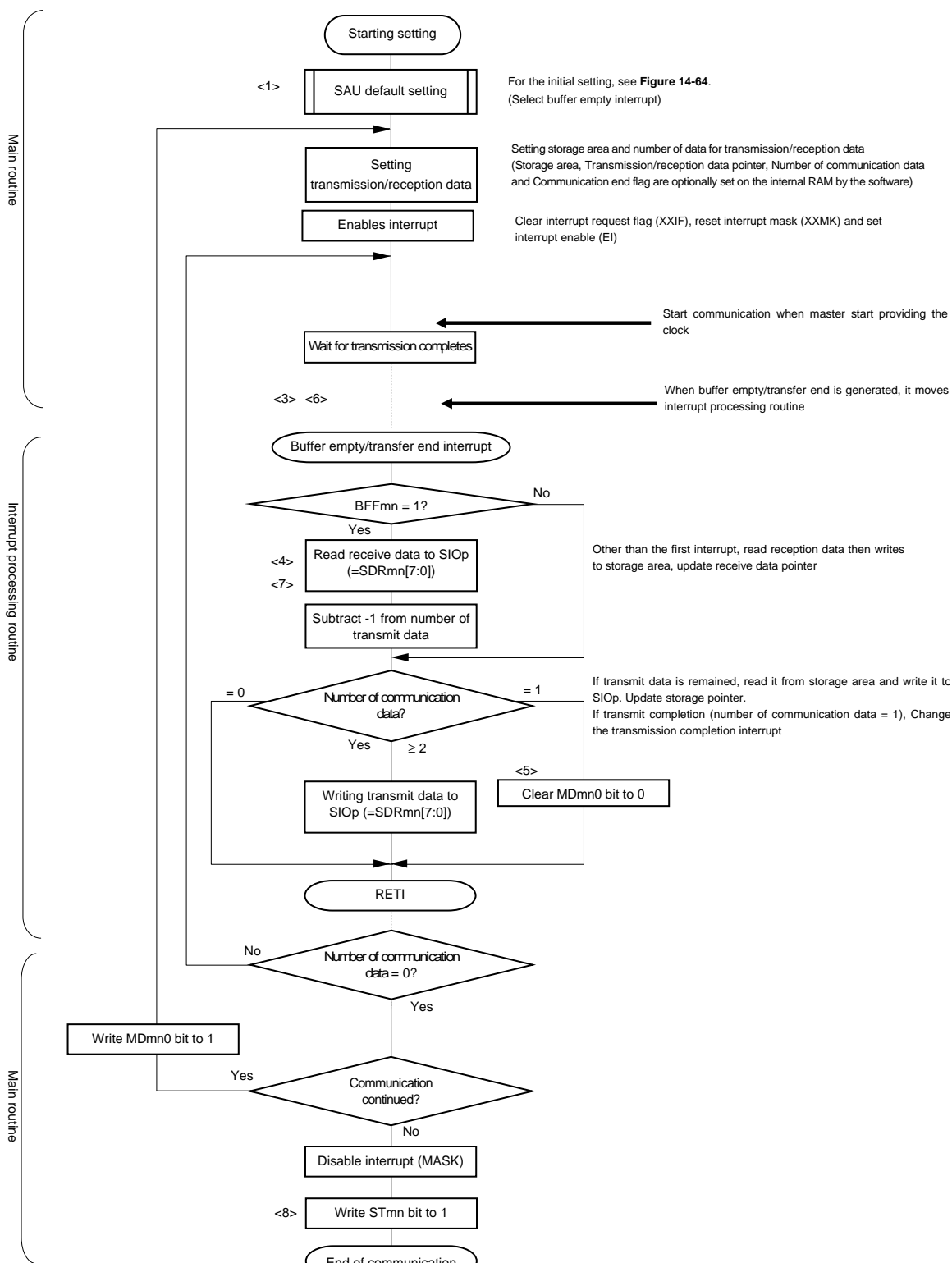


- Notes**
1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.
  2. The transmit data can be read by reading the SDRmn register during this period. 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 14-70 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.
  2. m: Unit number (m = 0), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10), mn = 00, 02

Figure 14-70. 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 14-69 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).



**14.5.7 SNOOZE mode function**

SNOOZE mode makes CSI operate reception by SCKp pin input detection while the STOP mode. Normally CSI stops communication in the STOP mode. But, using the SNOOZE mode makes reception CSI operate unless the CPU operation by detecting SCKp pin input. Only CSI00 can be set to the SNOOZE mode.

When using the CSI in SNOOZE mode, make the following setting before switching to the STOP mode (see **Figure 14-72 Flowchart of SNOOZE Mode Operation (once startup)** and **Figure 14-74 Flowchart of SNOOZE Mode Operation (continuous startup)**).

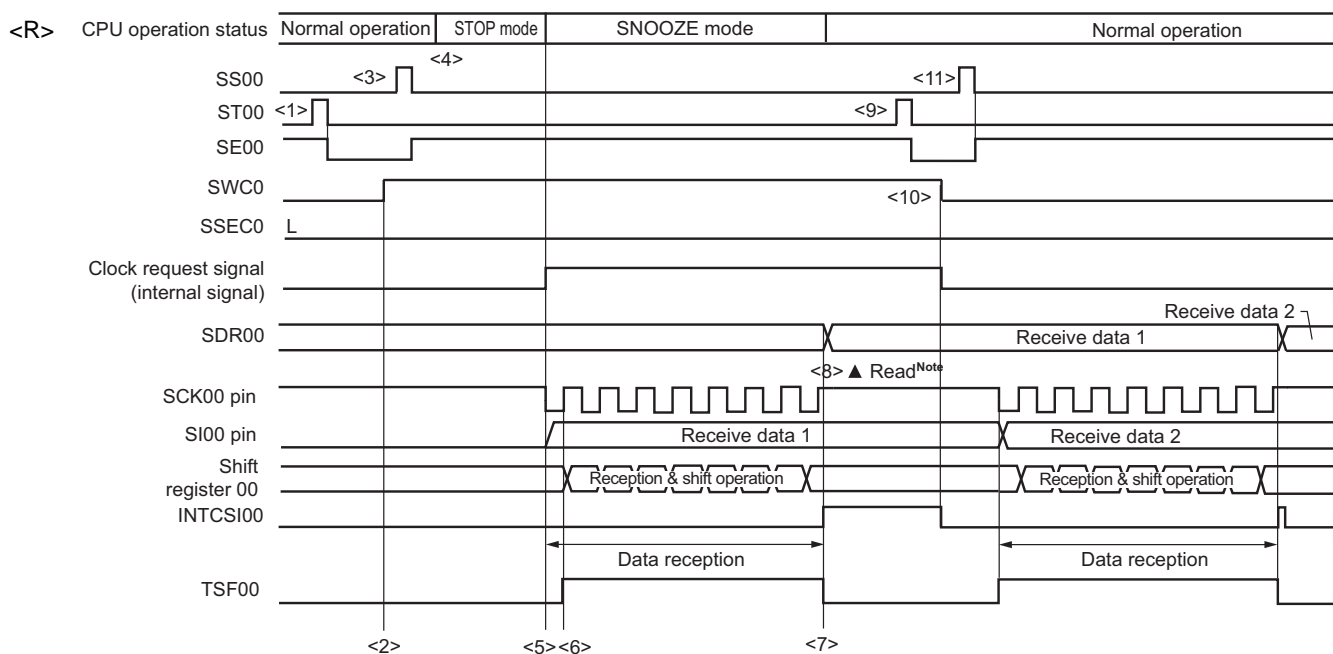
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has been completed, set the SSm0 bit of serial channel start register m (SSm) to 1.
- The CPU shifts to the SNOOZE mode on detecting the valid edge of the SCKp signal following a transition to the STOP mode. A CSIp starts reception on detecting input of the serial clock on the SCKp pin.

**Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for fCLK.**

**2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.**

**(1) SNOOZE mode operation (once startup)**

**Figure 14-71. 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).**

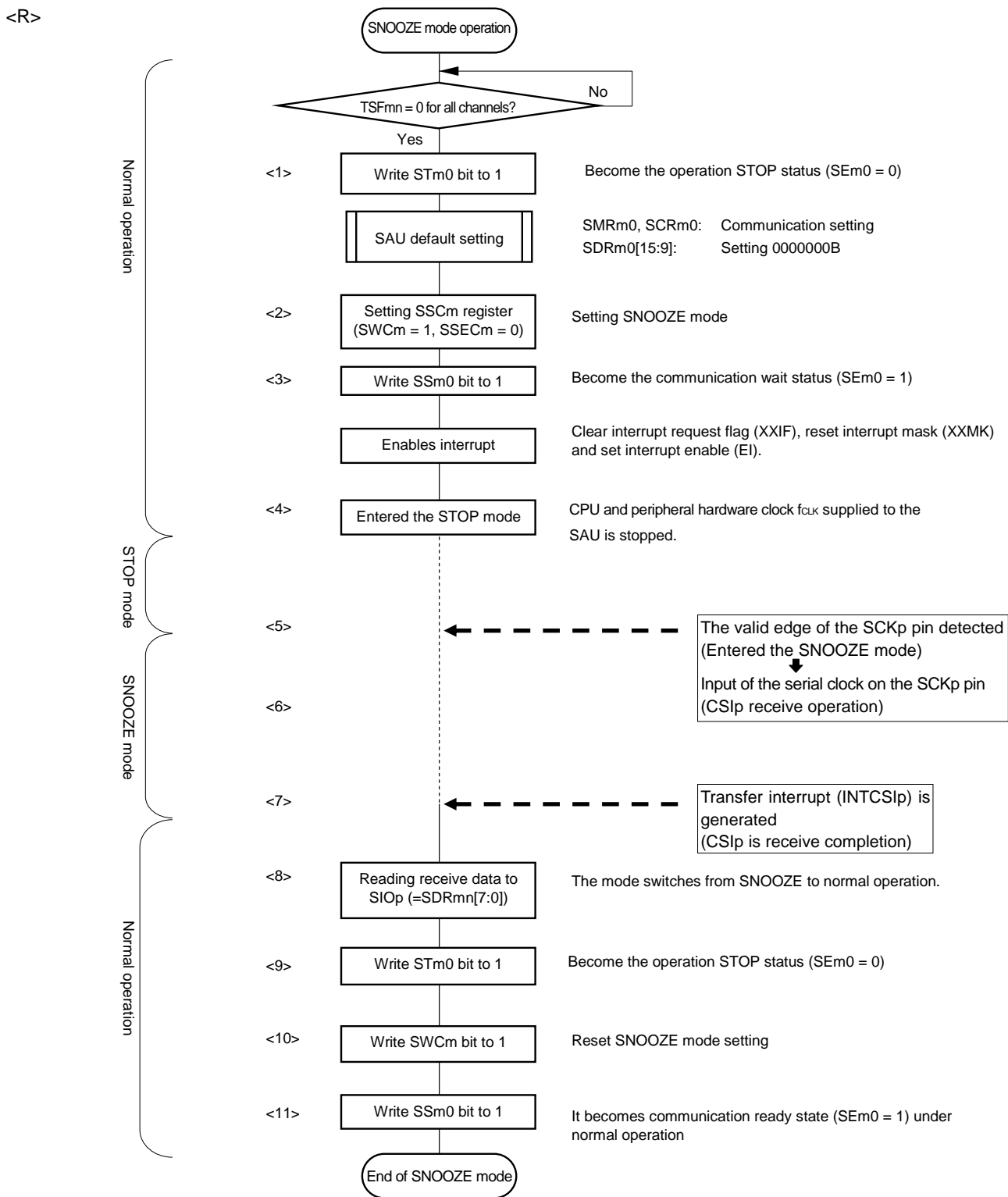
**And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).**

**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 14-72 Flowchart of SNOOZE Mode Operation (Once Startup).**

**2. m = 0; p = 00**

Figure 14-72. Flowchart of SNOOZE Mode Operation (Once Startup)



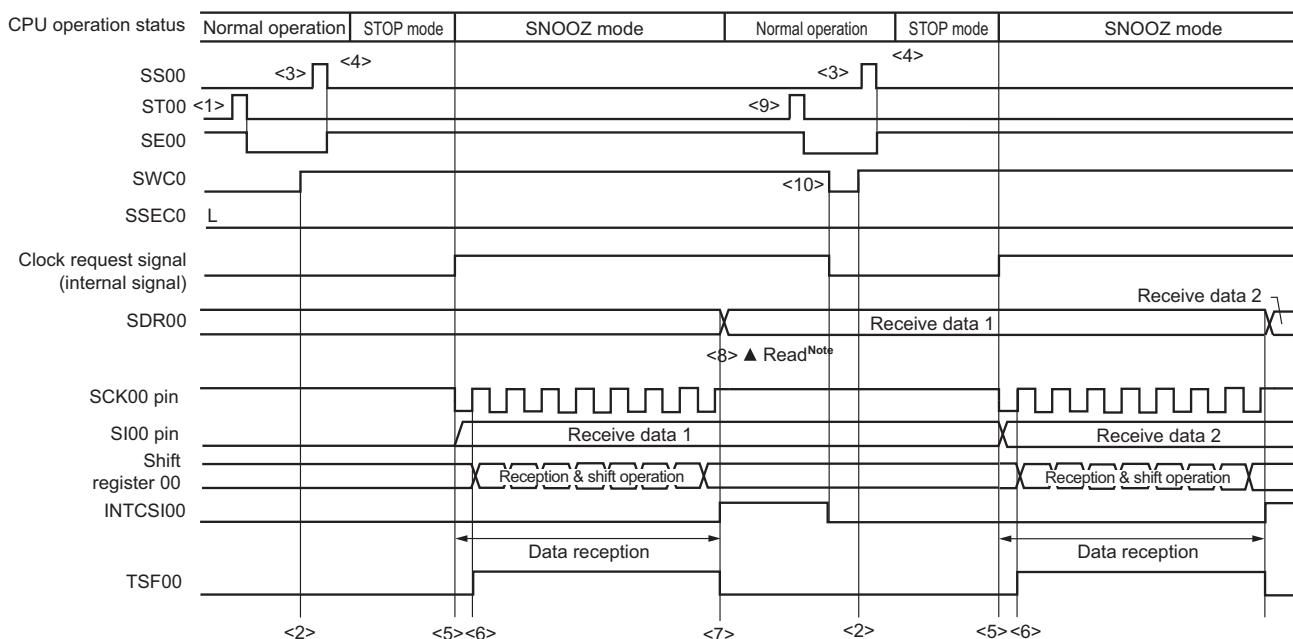
Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 14-71 Timing Chart of SNOOZE Mode Operation (once startup).

2. m = 0; p = 00

(2) SNOOZE mode operation (continuous startup)

Figure 14-73. Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)

<R>

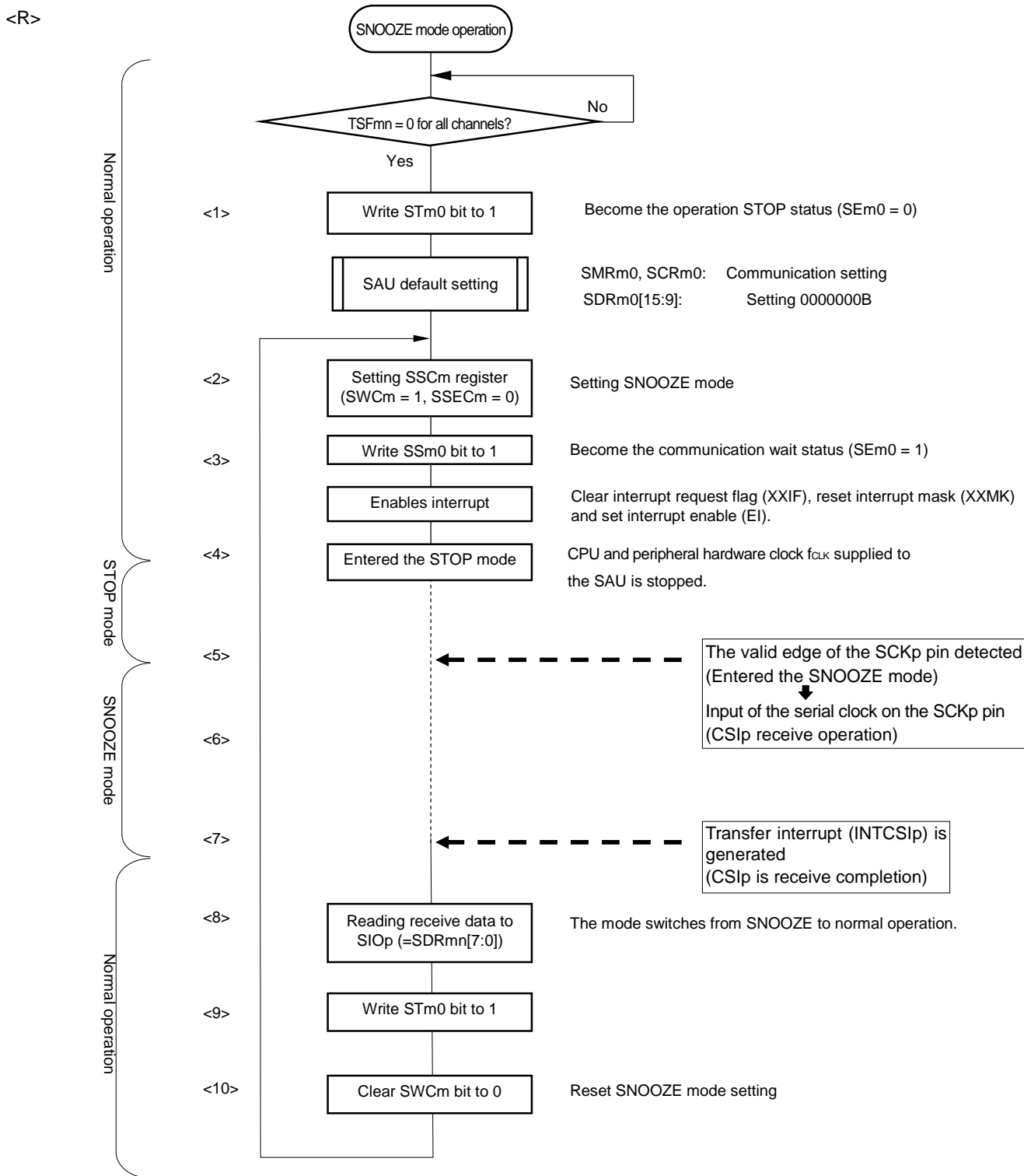


**Note** Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

- 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). And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE 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 14-74 Flowchart of SNOOZE Mode Operation (Continuous Startup).
  2. m = 0; p = 00

Figure 14-74. Flowchart of SNOOZE Mode Operation (Continuous Startup)



Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 14-73 Timing Chart of SNOOZE Mode Operation (Continuous Startup).

2. m = 0; p = 00

### 14.5.8 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI00, CSI10) communication can be calculated by the following expressions.

#### (1) Master

$$\text{(Transfer clock frequency)} = \{\text{Operation clock (f}_{MCK}\text{) frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2 \text{ [Hz]}$$

#### (2) Slave

$$\text{(Transfer clock frequency)} = \{\text{Frequency of serial clock (SCK) supplied by master}\}^{\text{Note}} \text{ [Hz]}$$

**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 14-2. Selection of Operation Clock For 3-Wire Serial I/O**

SMRmn Register	SPSm Register								Operation Clock (f <sub>CLK</sub> ) <sup>Note</sup>	
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	f <sub>CLK</sub> = 24 MHz
0	X	X	X	X	0	0	0	0	f <sub>CLK</sub>	24 MHz
	X	X	X	X	0	0	0	1	f <sub>CLK</sub> /2	12 MHz
	X	X	X	X	0	0	1	0	f <sub>CLK</sub> /2 <sup>2</sup>	6 MHz
	X	X	X	X	0	0	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	3 MHz
	X	X	X	X	0	1	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	1.5 MHz
	X	X	X	X	0	1	0	1	f <sub>CLK</sub> /2 <sup>5</sup>	750 kHz
	X	X	X	X	0	1	1	0	f <sub>CLK</sub> /2 <sup>6</sup>	375 kHz
	X	X	X	X	0	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	187.5 kHz
	X	X	X	X	1	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	93.8 kHz
	X	X	X	X	1	0	0	1	f <sub>CLK</sub> /2 <sup>9</sup>	46.9 kHz
	X	X	X	X	1	0	1	0	f <sub>CLK</sub> /2 <sup>10</sup>	23.4 kHz
	X	X	X	X	1	0	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	11.7 kHz
	X	X	X	X	1	1	0	0	f <sub>CLK</sub> /2 <sup>12</sup>	5.86 kHz
	X	X	X	X	1	1	0	1	f <sub>CLK</sub> /2 <sup>13</sup>	2.93 kHz
	X	X	X	X	1	1	1	0	f <sub>CLK</sub> /2 <sup>14</sup>	1.46 kHz
X	X	X	X	1	1	1	1	f <sub>CLK</sub> /2 <sup>15</sup>	732 Hz	
1	0	0	0	0	X	X	X	X	f <sub>CLK</sub>	24 MHz
	0	0	0	1	X	X	X	X	f <sub>CLK</sub> /2	12 MHz
	0	0	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>2</sup>	6 MHz
	0	0	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>3</sup>	3 MHz
	0	1	0	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>4</sup>	1.5 MHz
	0	1	0	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>5</sup>	750 kHz
	0	1	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>6</sup>	375 kHz
	0	1	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>7</sup>	187.5 kHz
	1	0	0	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>8</sup>	93.8 kHz
	1	0	0	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>9</sup>	46.9 kHz
	1	0	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>10</sup>	23.4 kHz
	1	0	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>11</sup>	11.7 kHz
	1	1	0	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>12</sup>	5.86 kHz
	1	1	0	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>13</sup>	2.93 kHz
	1	1	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>14</sup>	1.46 kHz
1	1	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>15</sup>	732 Hz	

**Note** When changing the clock selected for f<sub>CLK</sub> (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) 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, 2), mn = 00, 02

### 14.5.9 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI10) communication

The procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI10) communication is described in Figure 14-75.

**Figure 14-75. 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, 2), mn = 00, 02

## 14.6 Operation of UART (UART0 to UART3) 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 asynchronous communication UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can be implemented by using UART0, timer array unit 0 (channel 7), and an external interrupt (INTP0).

### [Data transmission/reception]

- Data length of 7, 8, or 9 bits<sup>Note</sup>
- Select the MSB/LSB first
- Level setting of transmit/receive data (selecting whether to reverse the level)
- Parity bit appending and parity check functions
- Stop bit appending, stop bit check function

### [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

In addition, UART0 and UART2 reception support the SNOOZE mode. When RxD pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only UART0, UART2 can be specified for the reception baud rate adjustment function.

The LIN-bus is accepted in UART0 (channels 0 and 1 of unit 0).

### [LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit 0 (channel 7)

**Note** Only UART0, UART2 can be specified for the 9-bit data length.



UART0 uses channels 0 and 1 of SAU0.

UART1 uses channels 2 and 3 of SAU0.

UART2 uses channels 0 and 1 of SAU1.

UART3 uses channels 2 and 3 of SAU1.

• 64-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I <sup>2</sup> C	
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00	
	1	–		–	
	2	CSI10		UART1	IIC10
	3	–		–	
1	0	–	UART2	–	
	1	–		–	

• 80-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I <sup>2</sup> C	
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00	
	1	–		–	
	2	CSI10		UART1	IIC10
	3	–		–	
1	0	–	UART2	–	
	1	–		–	
	2	–	UART3	–	
	3	–		–	

Select any function for each channel. Only the selected function is possible. If UART0 is selected for channels 0 and 1 of unit 0, for example, these channels cannot be used for CSI00. At this time, however, channel 2, 3, or other channels of the same unit can be used for a function other than UART0, such as CSI10, UART1, and IIC10.

**Caution** When using a serial array unit for UART, both the transmitter side (even-numbered channel) and the receiver side (odd-numbered channel) can only be used for UART.

UART performs the following four types of communication operations.

- UART transmission (See 14.6.1.)
- UART reception (See 14.6.2.)
- LIN transmission (UART2 only) (See 14.7.1.)
- LIN reception (UART2 only) (See 14.7.2.)

### 14.6.1 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	UART1	UART2	UART3
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	TxD0	TxD1	TxD2	TxD3
Interrupt	INTST0	INTST1	INTST2	INTST3
	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 <sup>Note 1</sup>			
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR <sub>mn</sub> [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] <sup>Note 2</sup>			
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)			
Parity bit	The following selectable <ul style="list-style-type: none"> <li>• No parity bit</li> <li>• Appending 0 parity</li> <li>• Appending even parity</li> <li>• Appending odd parity</li> </ul>			
Stop bit	The following selectable <ul style="list-style-type: none"> <li>• Appending 1 bit</li> <li>• Appending 2 bits</li> </ul>			
Data direction	MSB or LSB first			

**Notes 1.** Only UART0, UART2 can be specified for the 9-bit data length.

**2.** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)**).

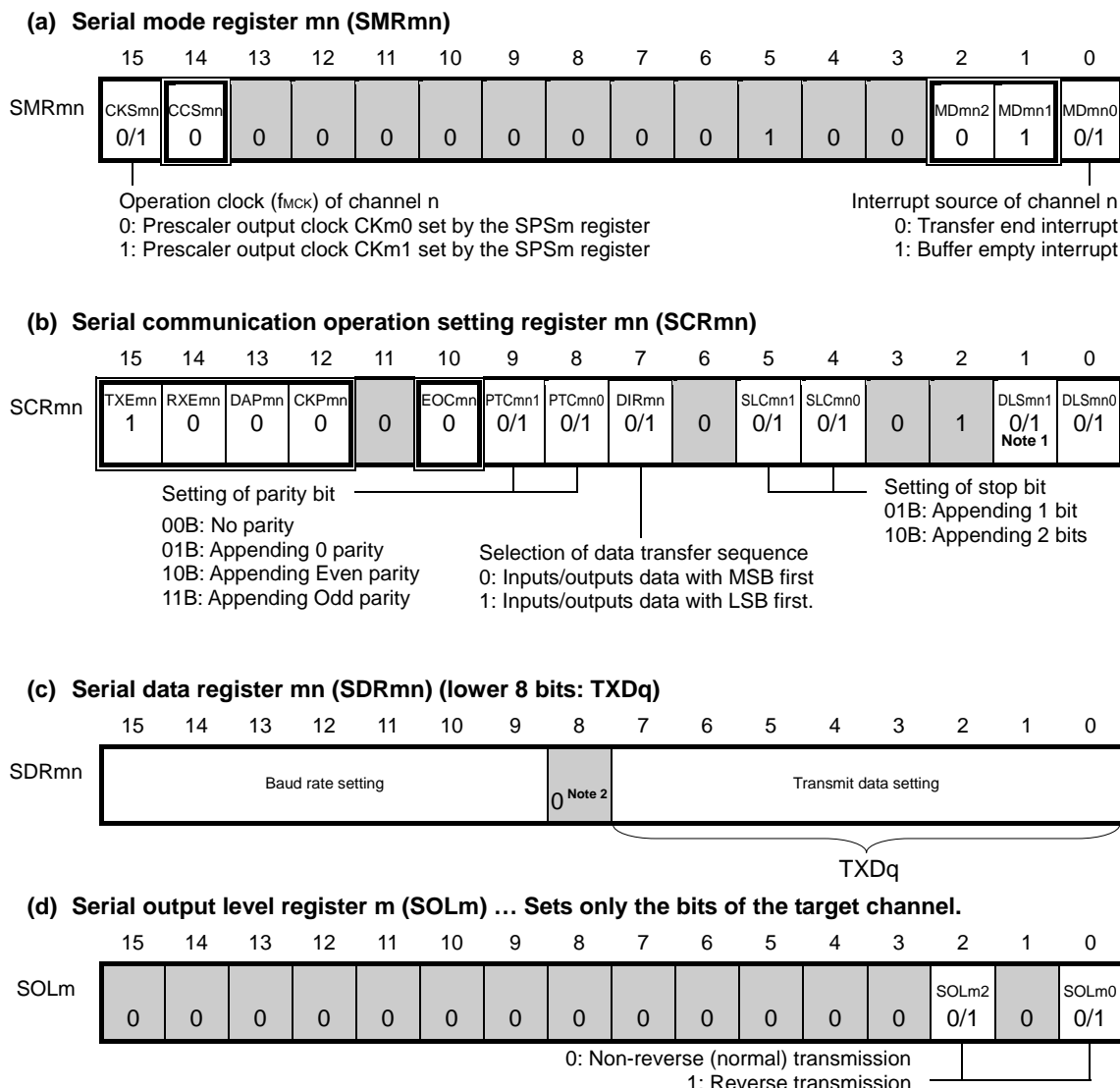
**Remarks 1.**  $f_{MCK}$ : Operation clock frequency of target channel

$f_{CLK}$ : System clock frequency

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

(1) Register setting

Figure 14-76. Example of Contents of Registers for UART Transmission of UART (UART0 to UART3) (80-pin products) (1/2)



- Notes**
- Only provided for the SCR00, SCR01, SCR10 and SCR11 registers. This bit is fixed to 1 for the other registers.
  - When UART0 performs 9-bit communication (by setting the DLS001 and DLS000 bits of the SCR00 register to 1), bits 0 to 8 of the SDR00 register are used as the transmission data specification area. Only UART0, UART2 can be specified for the 9-bit data length.

- Remarks**
- m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3), mn = 00, 02, 10, 12
  - : 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 14-76. Example of Contents of Registers for UART Transmission of UART (UART0 to UART3) (80-pin products) (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	1	CKOm2 Note 2 ×	1	CKOm0 Note 2 ×	0	0	0	0	1	SOm2 0/1 Note 1	1	SOm0 0/1 Note 1

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	SOEm2 0/1	0	SOEm0 0/1

**(g) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.**

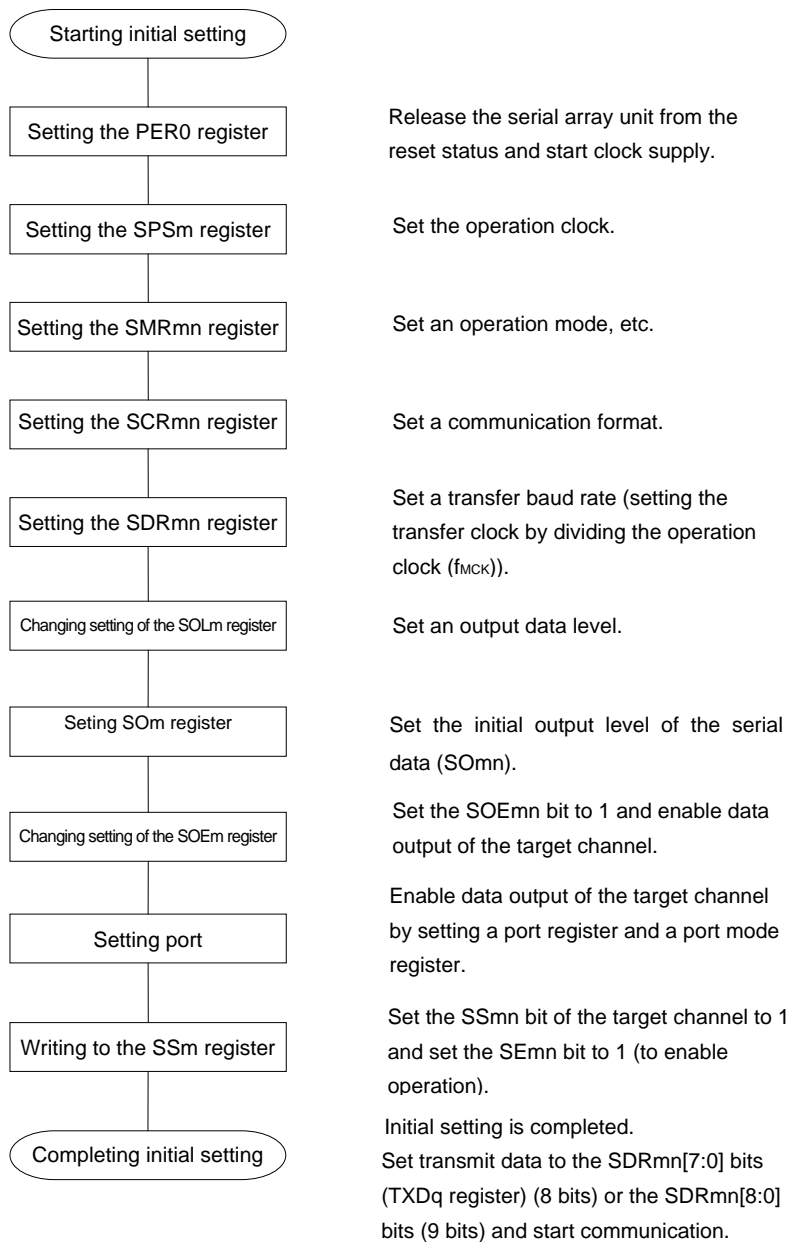
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

- Notes**
1. 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.
  2. Serial array unit 0 only.

- Remarks**
1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)  
mn = 00, 02, 10, 12
  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 14-77. Initial Setting Procedure for UART Transmission



**Figure 14-78. Procedure for Stopping UART Transmission**

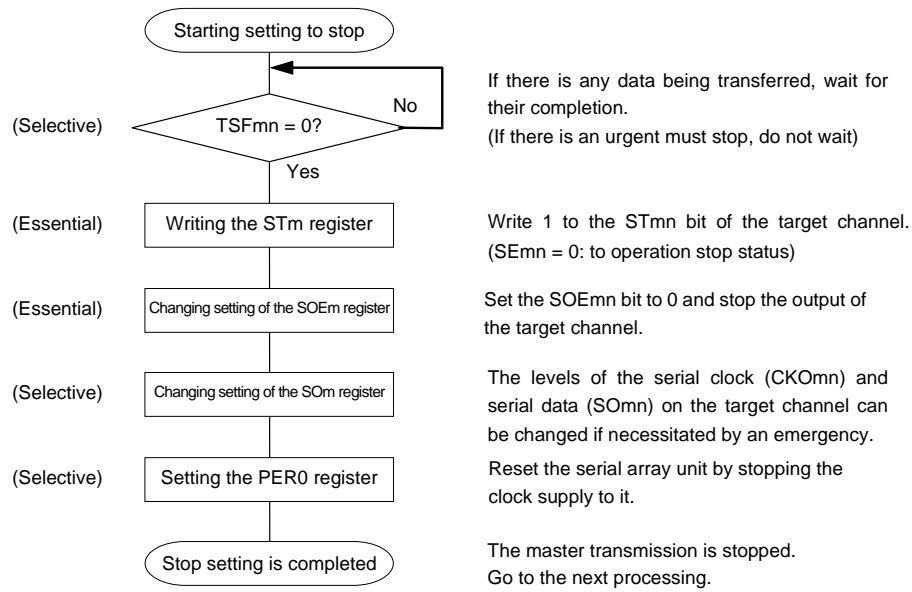
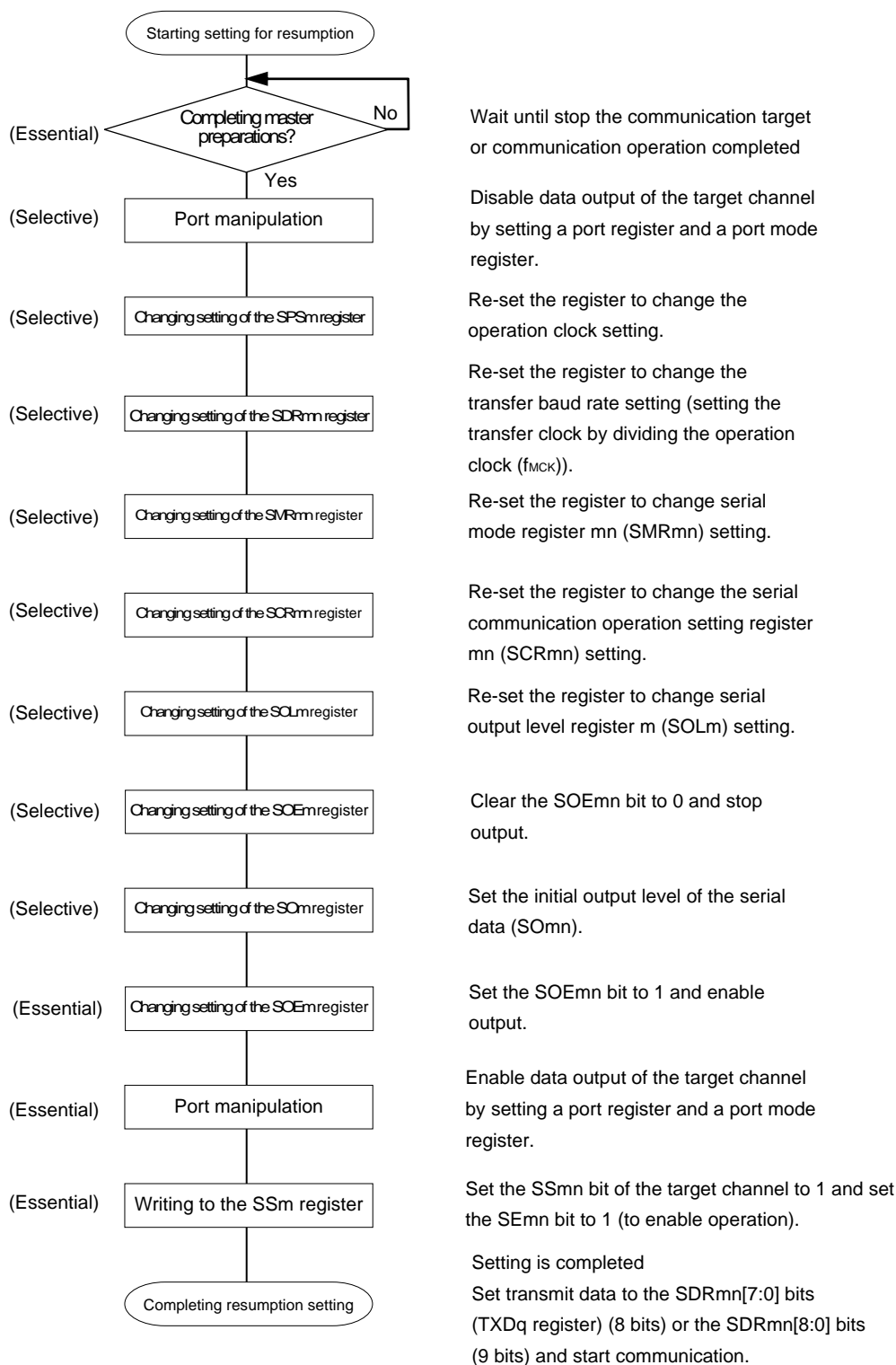


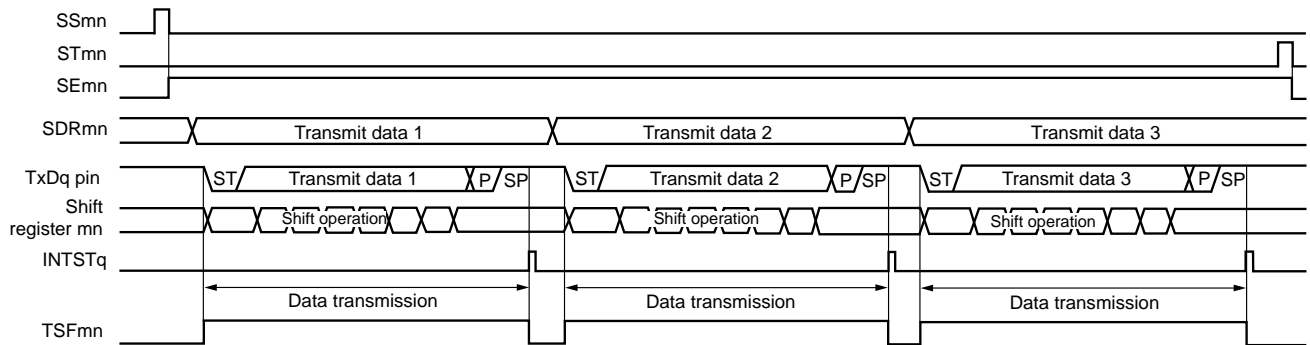
Figure 14-79. Procedure for Resuming UART Transmission



**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)

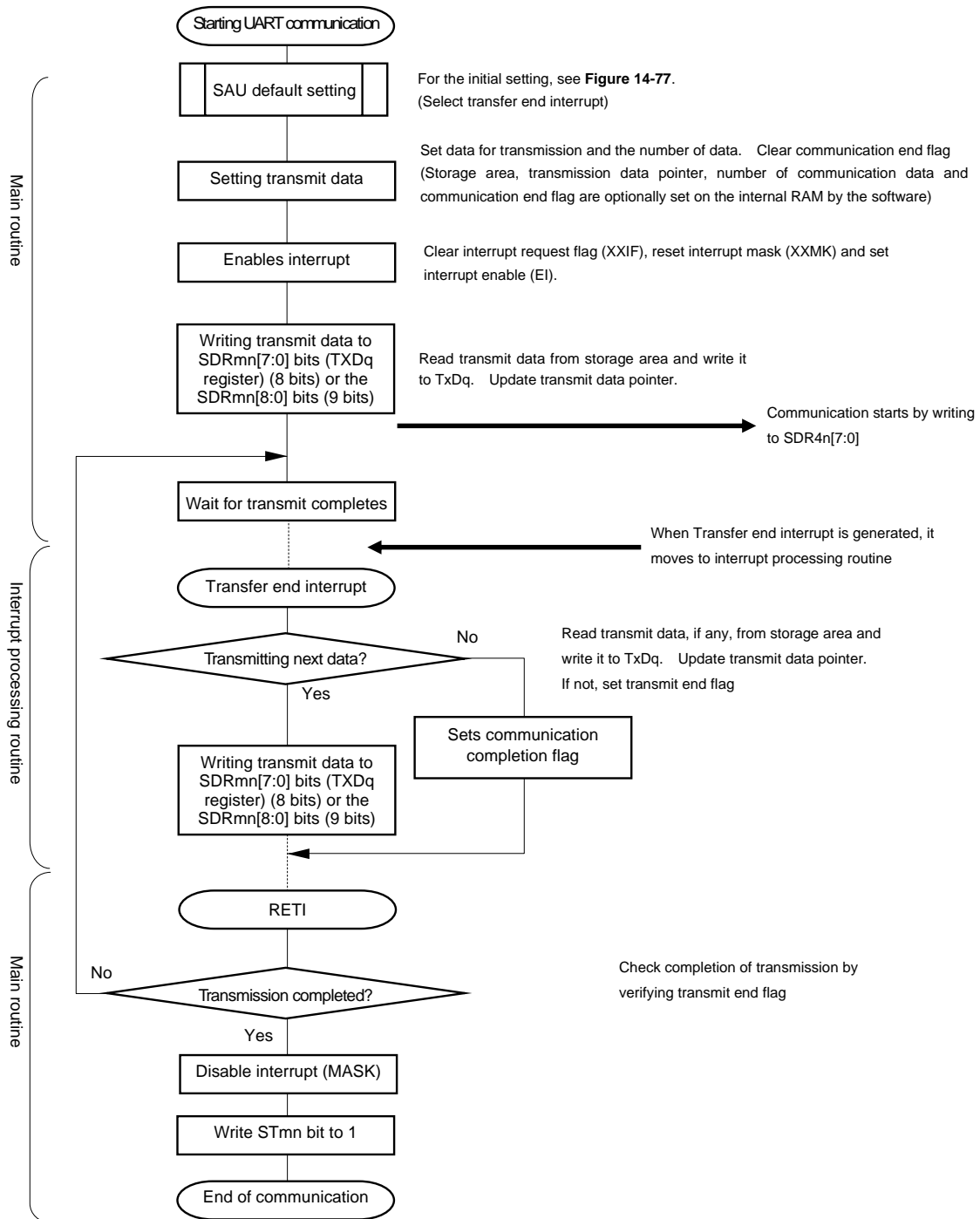
Figure 14-80. Timing Chart of UART Transmission (in Single-Transmission Mode)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)  
 mn = 00, 02, 10, 12

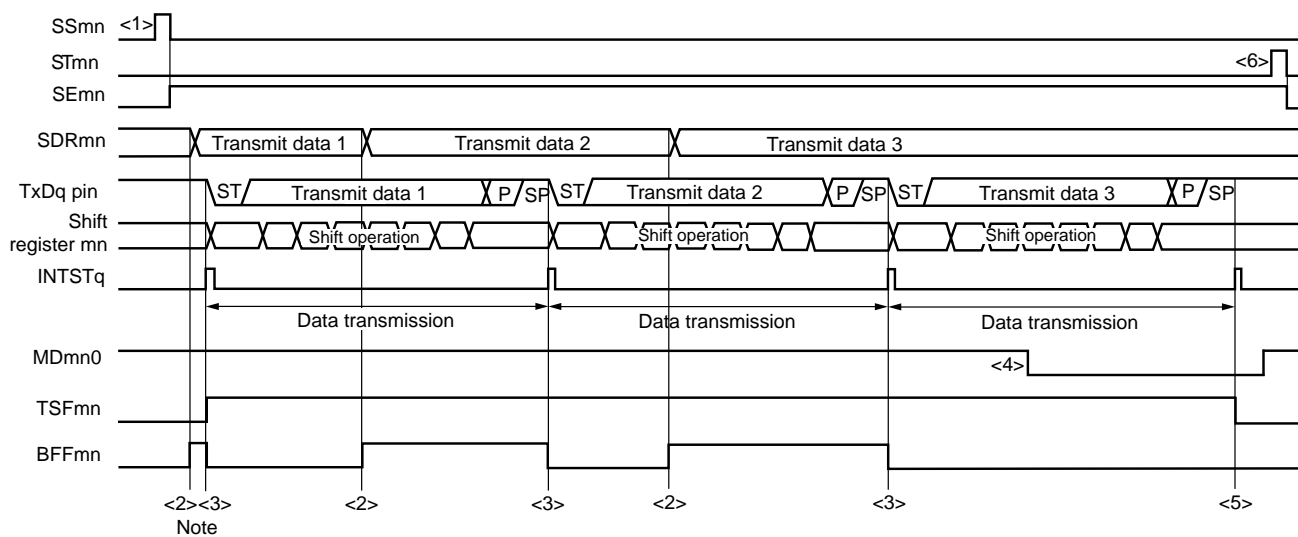


Figure 14-81. Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 14-82. Timing Chart of UART Transmission (in Continuous Transmission Mode)

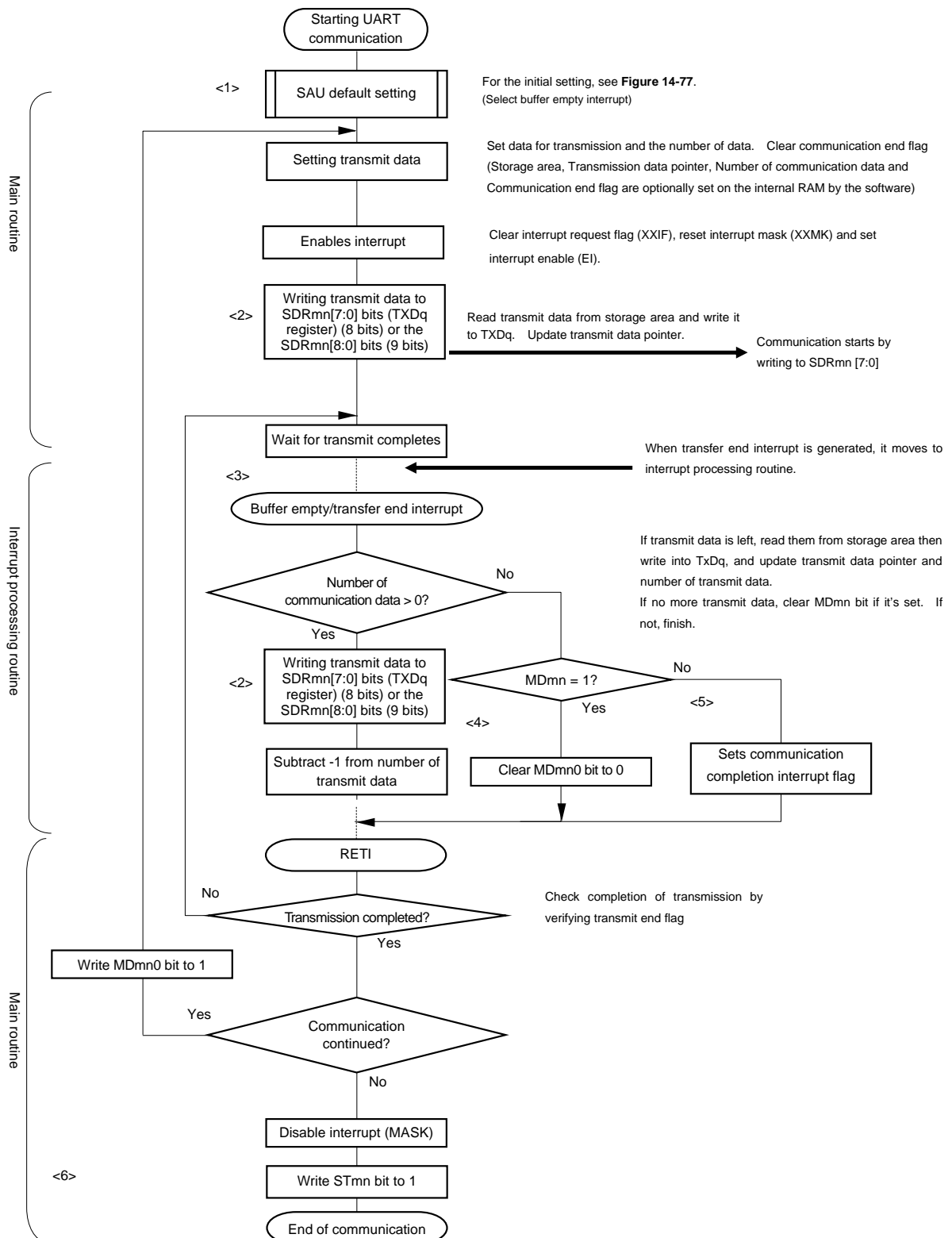


**Note** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

**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, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)  
 mn = 00, 02, 10, 12

Figure 14-83. Flowchart of UART Transmission (in Continuous Transmission Mode)



**Remark** <1> to <6> in the figure correspond to <1> to <6> in Figure 14-82 Timing Chart of UART Transmission (in Continuous Transmission Mode).

### 14.6.2 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	UART1	UART2	UART3
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1	Channel 3 of SAU1
Pins used	RxD0	RxD1	RxD2	RxD3
Interrupt	INTSR0	INTSR1	INTSR2	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	INTSRE0	INTSRE1	INTSRE2	INTSRE3
Error detection flag	<ul style="list-style-type: none"> <li>• Framing error detection flag (FEFmn)</li> <li>• Parity error detection flag (PEFmn)</li> <li>• Overrun error detection flag (OVFmn)</li> </ul>			
Transfer data length	7, 8 or 9 bits <sup>Note 1</sup>			
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] <sup>Note 2</sup>			
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)			
Parity bit	The following selectable <ul style="list-style-type: none"> <li>• No parity bit (no parity check)</li> <li>• No parity judgment (0 parity)</li> <li>• Even parity check</li> <li>• Odd parity check</li> </ul>			
Stop bit	Appending 1 bit			
Data direction	MSB or LSB first			

**Notes 1.** Only UART0, UART2 can be specified for the 8-bit data length.

**2.** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)**).

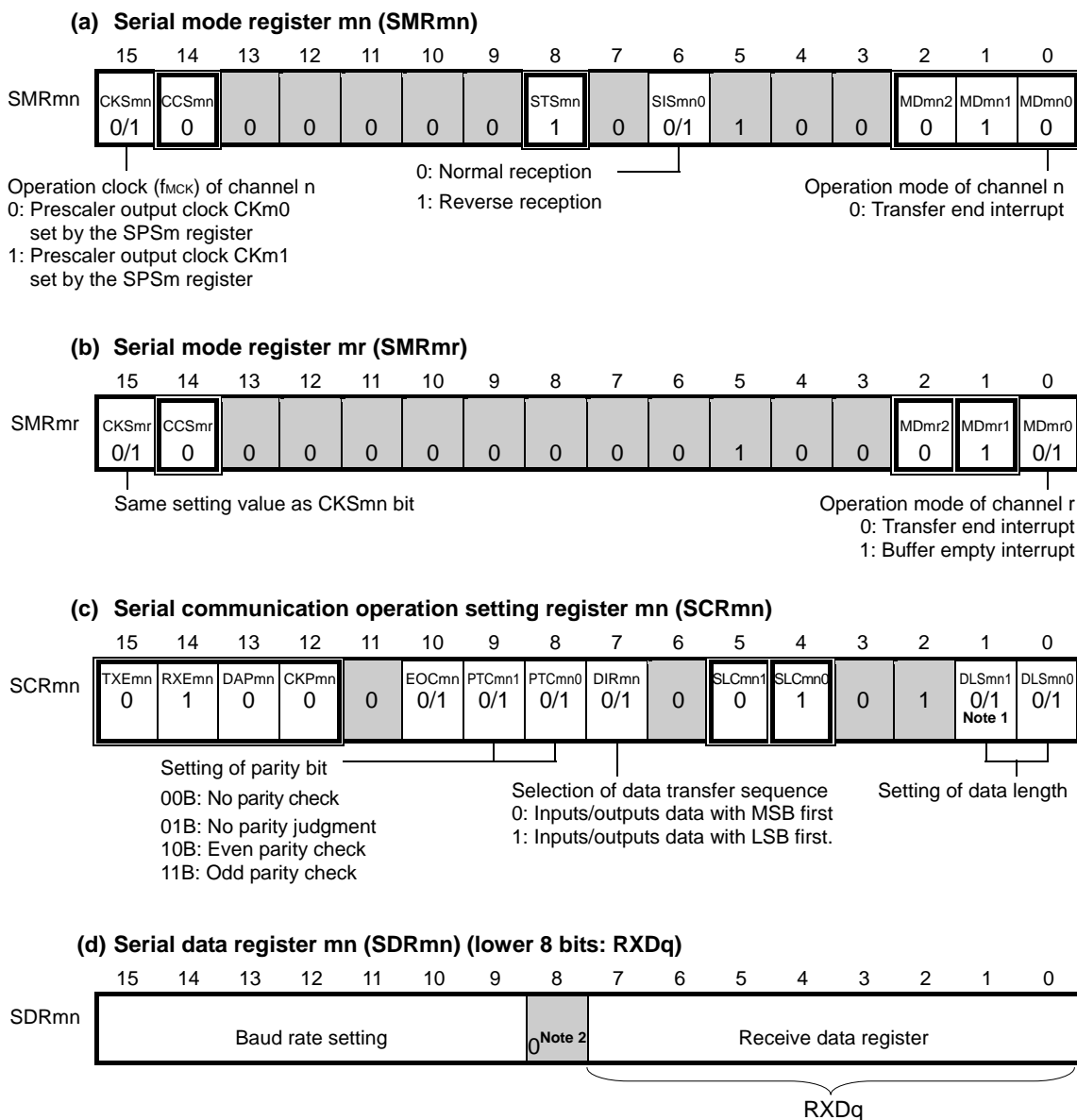
**Remarks 1.**  $f_{MCK}$ : Operation clock frequency of target channel

$f_{CLK}$ : System clock frequency

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13

(1) Register setting

Figure 14-84. Example of Contents of Registers for UART Reception of UART (UART0 to UART3) (80-pin products) (1/2)



**Notes 1.** Only provided for the SCR00, SCR01, SCR10 and SCR11 registers. This bit is fixed to 1 for the other registers.

**2.** When UART performs 9-bit communication, bits 0 to 8 of the SDRm1 register are used as the transmission data specification area. Only UART0, UART2 can be specified for the 8-bit data length.

**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, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13  
 r: Channel number (r = n - 1), q: UART number (q = 0 to 3)

**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 14-84. Example of Contents of Registers for UART Reception of UART (UART0 to UART3) (80-pin products) (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	1	CKOm2 Note ×	1	CKOm0 Note ×	0	0	0	0	1	SOm2 ×	1	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	SOEm2 ×	0	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	SSm3 0/1	SSm2 ×	SSm1 0/1	SSm0 ×

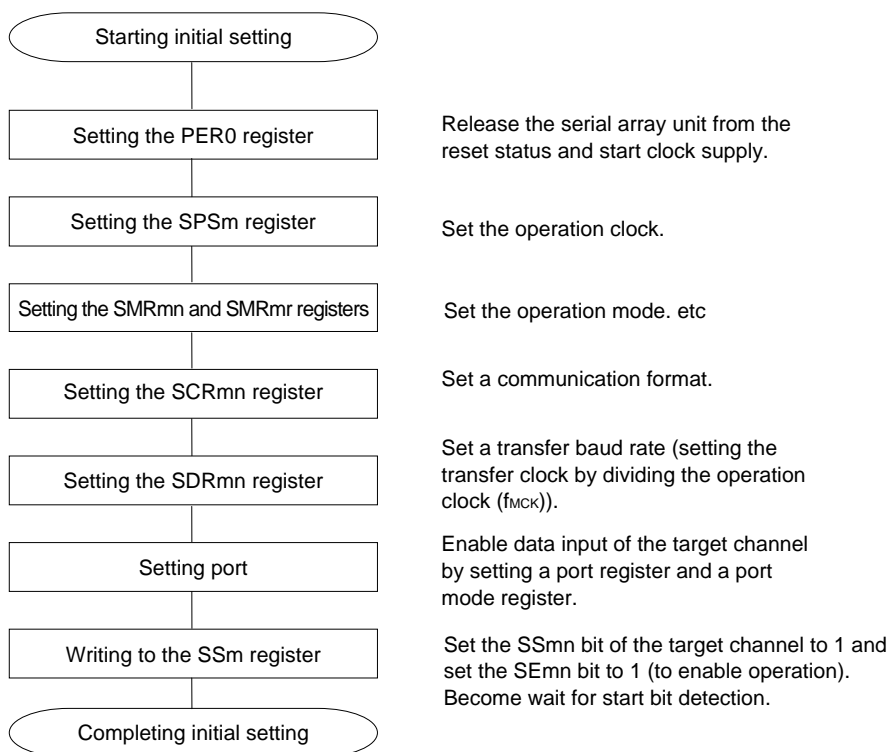
**Note** Serial array unit 0 only.

**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**
- m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13  
r: Channel number (r = n – 1), q: UART number (q = 0 to 3)
  - ☐: 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 14-85. Initial Setting Procedure for UART Reception



**Caution** Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more  $f_{MCK}$  clocks have elapsed.

Figure 14-86. Procedure for Stopping UART Reception

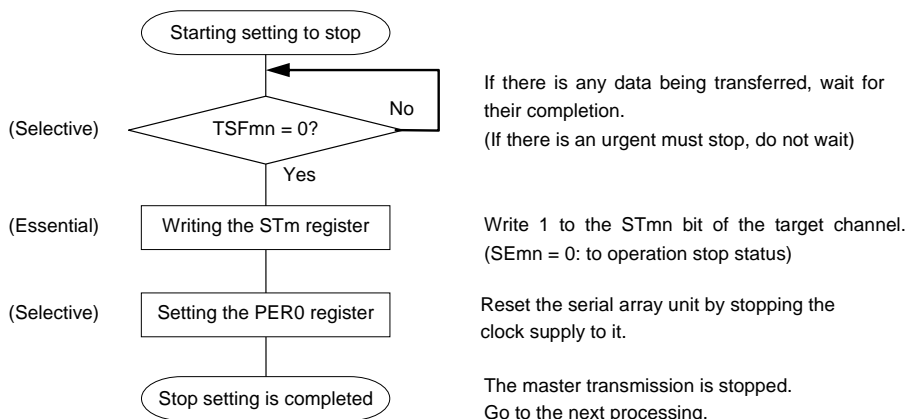
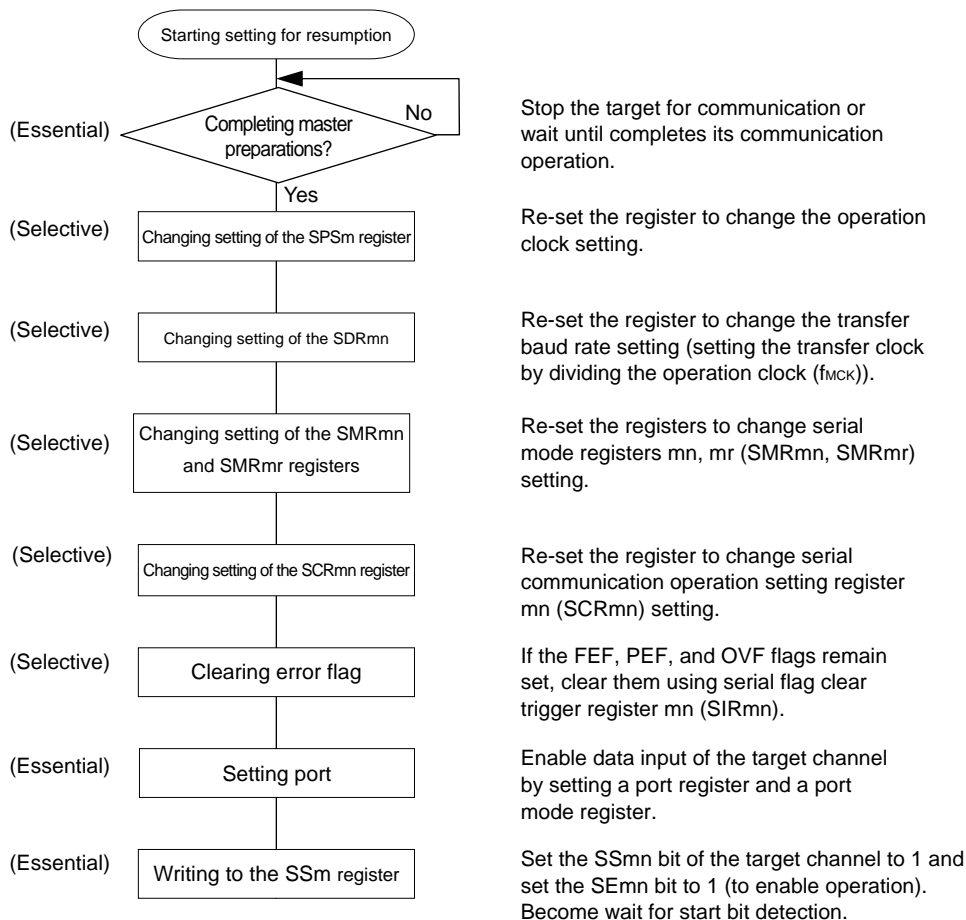


Figure 14-87. Procedure for Resuming UART Reception



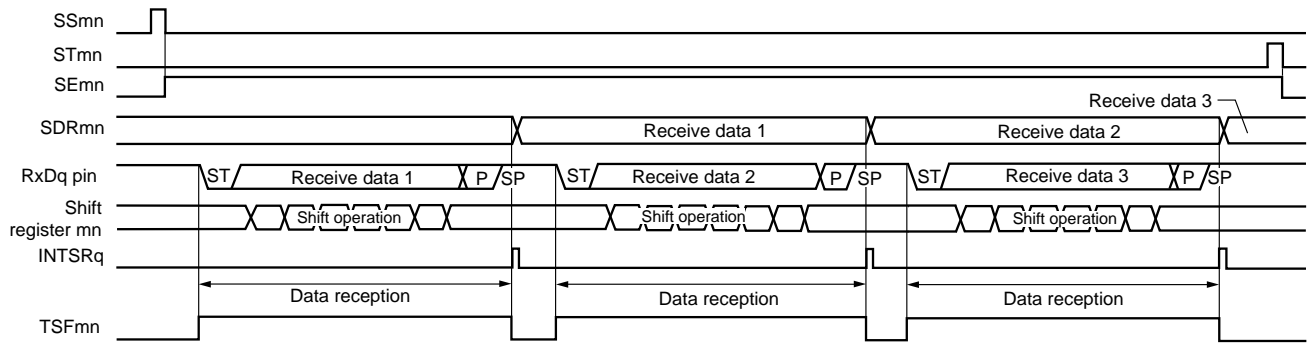
**Caution** After is set RXEmn bit to 1 of SCRmn register, set the SSmn = 1 from an interval of at least four clocks of f<sub>MCK</sub>.

**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.



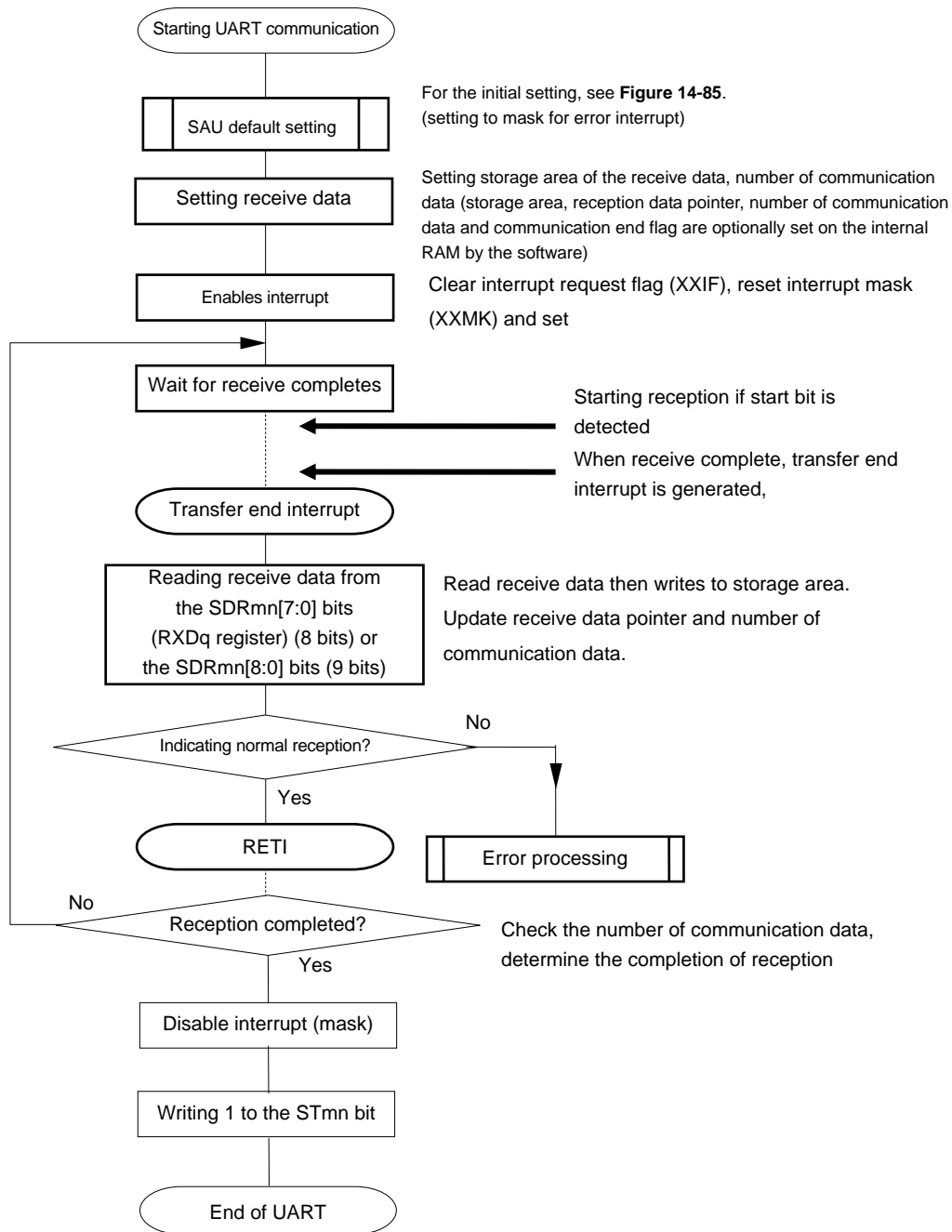
(3) Processing flow

Figure 14-88. Timing Chart of UART Reception



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13  
 r: Channel number (r = n - 1), q: UART number (q = 0 to 3)

Figure 14-89. Flowchart of UART Reception



### 14.6.3 SNOOZE mode function

SNOOZE mode makes UART operate reception by RxDq pin input detection while the STOP mode. Normally the UART stops communication in the STOP mode. However, using the SNOOZE mode enables the UART to perform reception operations without CPU operation. Only UART0 and UART2 can be set to the SNOOZE mode.

When using UARTq in the SNOOZE mode, make the following settings before entering the STOP mode. (See **Figure 14-92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)** and **Figure 14-94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).**)

- In the SNOOZE mode, the baud rate setting for UART reception needs to be changed to a value different from that in normal operation. Set the SPSm register and bits 15 to 9 of the SDRmn register with reference to Table 14-3.
- Set the EOCmn and SSECmn bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.
- <R> • When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has completed, set the SSm0 bit of serial channel start register m (SSm) to 1.
- <R> • A UARTq starts reception in SNOOZE mode on detecting input of the start bit on the RxDq pin following a transition of the CPU to the STOP mode.

**Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock ( $f_{IH}$ ) is selected for  $f_{CLK}$ .**

**2. The transfer rate in the SNOOZE mode is only 4800 bps.**

**3. When SWCm = 1, UARTq can be used only when the reception operation is started in the STOP mode. When used simultaneously with another SNOOZE mode function or interrupt, if the reception operation is started in a state other than the STOP mode, such as those given below, data may not be received correctly and a framing error or parity error may be generated.**

- When after the SWCm bit has been set to 1, the reception operation is started before the STOP mode is entered
- When the reception operation is started while another function is in the SNOOZE mode
- When after returning from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWCm bit is returned to 0

**4. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFmn, FEFmn, or OVFmn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFmn, FEFmn, or OVFmn flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.**

- <R> **5. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxDq signal. Note, however, that transfer through the UART channel may not start and the CPU may remain in the SNOOZE mode if an input pulse on the RxDq pin is too short to be detected as a start bit. In such cases, data may not be received correctly, and this may lead to a framing error or parity error in the next UART transfer.**

**Table 14-3. Baud Rate Setting for UART Reception in SNOOZE Mode**

High-speed On-chip Oscillator ( $f_{IH}$ )	Baud Rate for UART Reception in SNOOZE Mode			
	Baud Rate of 4800 bps			
	Operation Clock ( $f_{MCK}$ )	SDRmn[15:9]	Maximum Permissible Value	Minimum Permissible Value
24 MHz $\pm$ 1.0% <sup>Note</sup>	$f_{CLK}/2^5$	79	1.60%	-2.18%
16 MHz $\pm$ 1.0% <sup>Note</sup>	$f_{CLK}/2^4$	105	2.27%	-1.53%
12 MHz $\pm$ 1.0% <sup>Note</sup>	$f_{CLK}/2^4$	79	1.60%	-2.19%
8 MHz $\pm$ 1.0% <sup>Note</sup>	$f_{CLK}/2^3$	105	2.27%	-1.53%
6 MHz $\pm$ 1.0% <sup>Note</sup>	$f_{CLK}/2^3$	79	1.60%	-2.19%
4 MHz $\pm$ 1.0% <sup>Note</sup>	$f_{CLK}/2^2$	105	2.27%	-1.53%
3 MHz $\pm$ 1.0% <sup>Note</sup>	$f_{CLK}/2^2$	79	1.60%	-2.19%
2 MHz $\pm$ 1.0% <sup>Note</sup>	$f_{CLK}/2$	105	2.27%	-1.54%
1 MHz $\pm$ 1.0% <sup>Note</sup>	$f_{CLK}$	105	2.27%	-1.57%

**Note** When the accuracy of the clock frequency of the high-speed on-chip oscillator is  $\pm 1.5\%$  or  $\pm 2.0\%$ , the permissible range becomes smaller as shown below.

- In the case of  $f_{IH} \pm 1.5\%$ , perform (Maximum permissible value - 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of  $f_{IH} \pm 2.0\%$ , perform (Maximum permissible value - 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

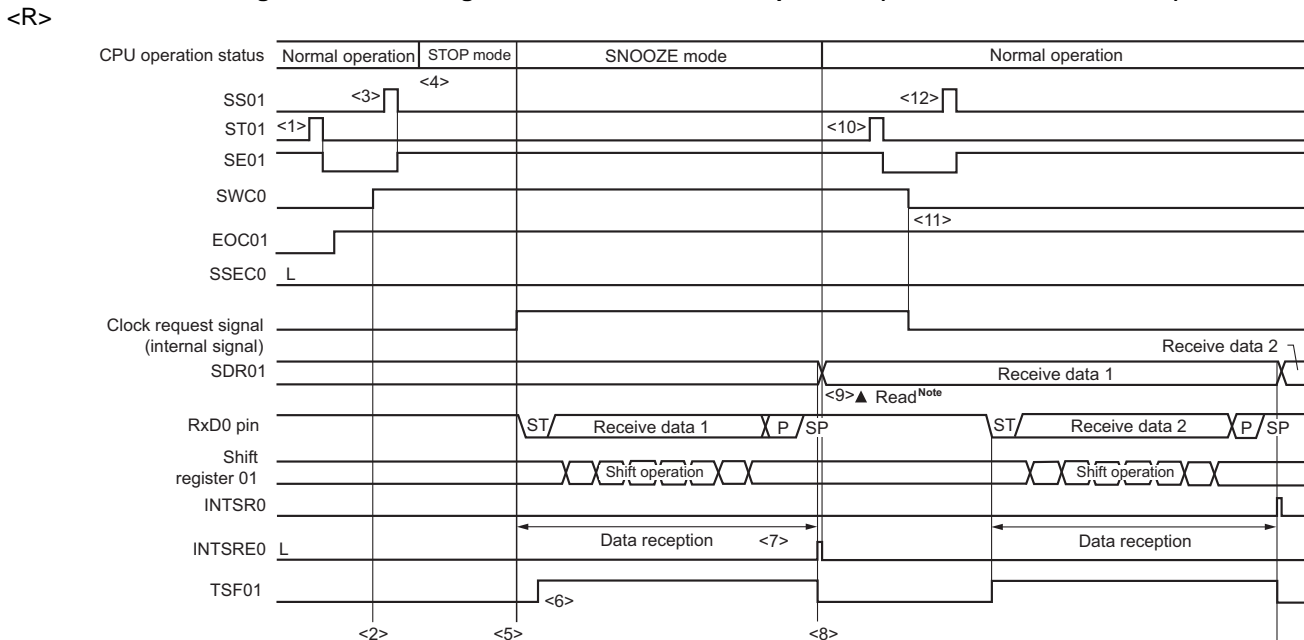
**Remark** The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception.

The baud rate on the transmitting side should be set to fall inside this range.

**(1) SNOOZE mode operation (EOCm1 = 0, SSECm = 0/1)**

Because of the setting of EOCm1 = 0, even though a communication error occurs, an error interrupt (INTSREq) is not generated, regardless of the setting of the SSECm bit. A transfer end interrupt (INTSRq) will be generated.

**Figure 14-90. Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)**



**Note** Read the received data when SWCm is 1

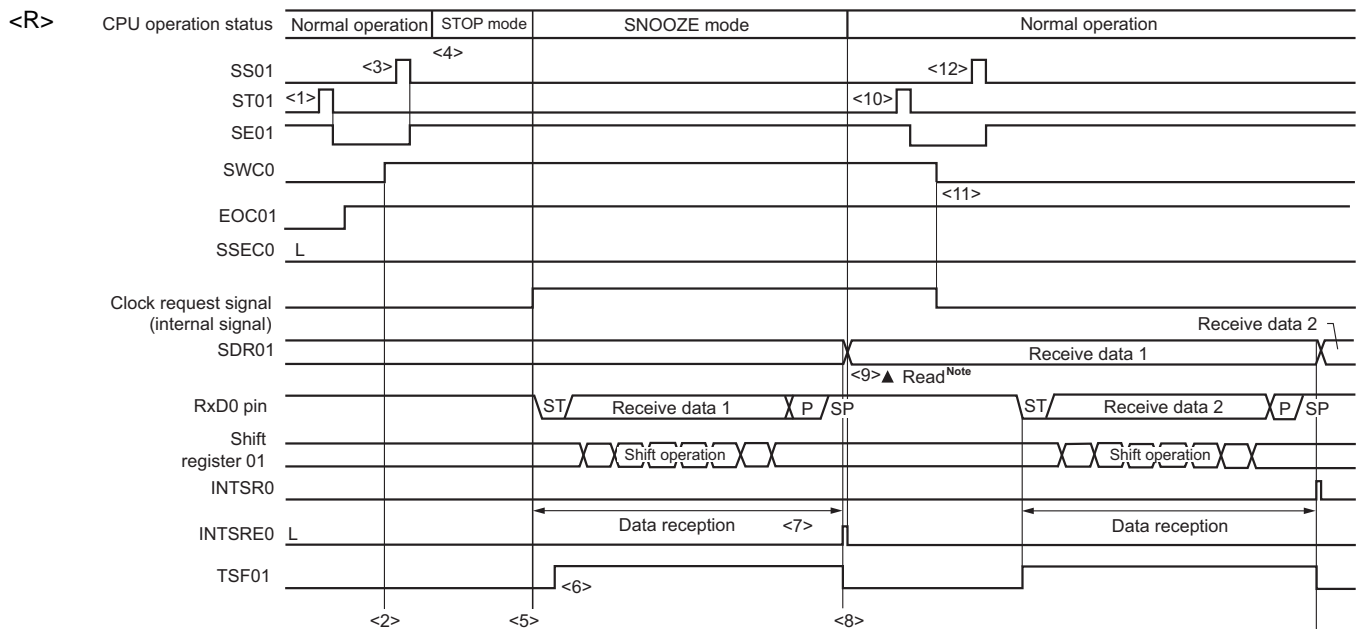
**Caution** Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEM1 bit, and stop the operation).  
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

**Remarks 1.** <1> to <12> in the figure correspond to <1> to <12> in **Figure 14-92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**..  
**2.** m = 0, 1; q = 0, 2

**(2) SNOOZE mode operation (EOCm1 = 1, SSECm = 0: Error interrupt (INTSREq) generation is enabled)**

Because EOCm1 = 1 and SSECm = 0, an error interrupt (INTSREq) is generated when a communication error occurs.

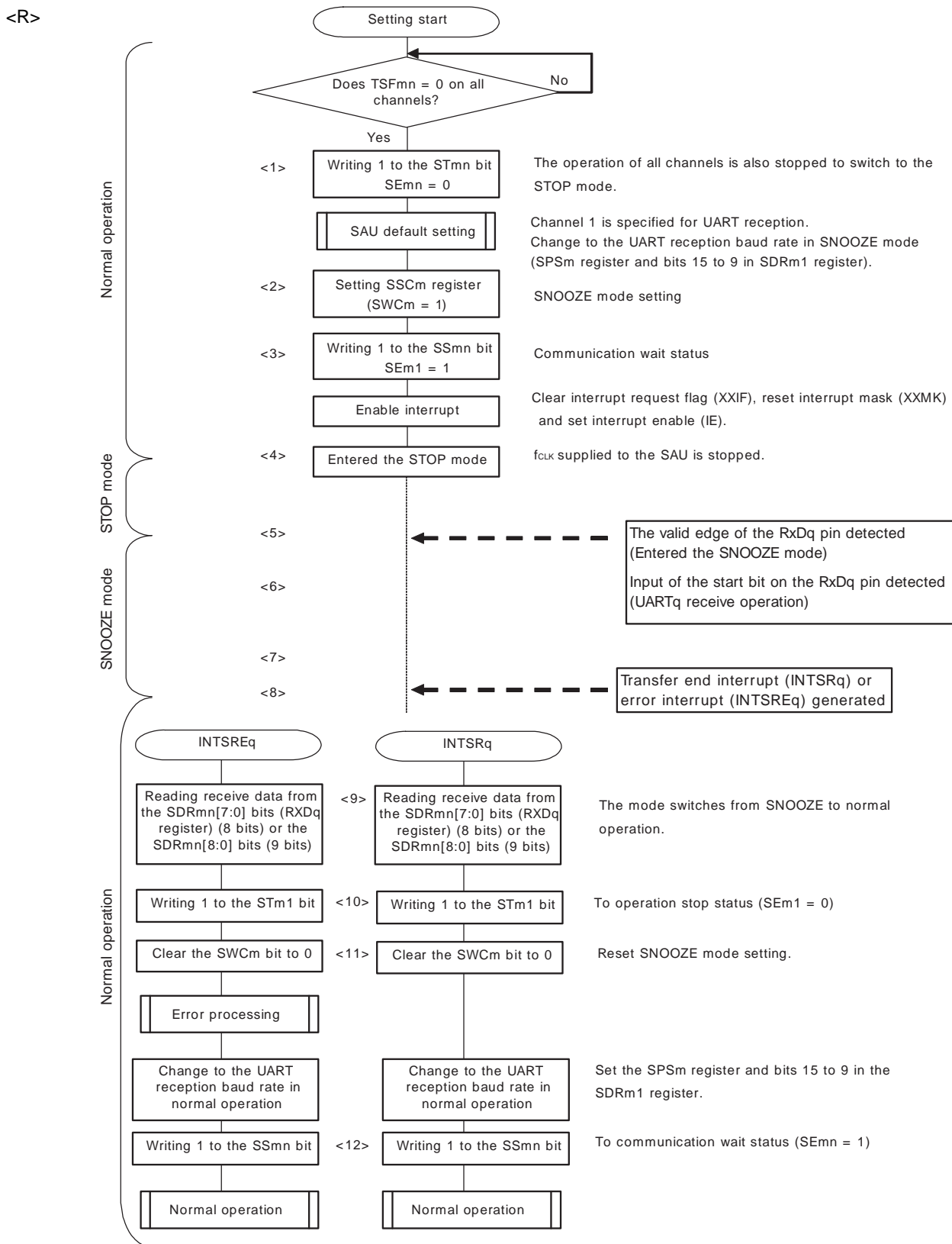
**Figure 14-91. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)**



**Caution** Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation). And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

- Remarks**
1. <1> to <12> in the figure correspond to <1> to <12> in Figure 14-92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).
  2. m = 0, 1; q = 0, 2

Figure 14-92. Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECM = 0/1 or EOCm1 = 1, SSECM = 0)



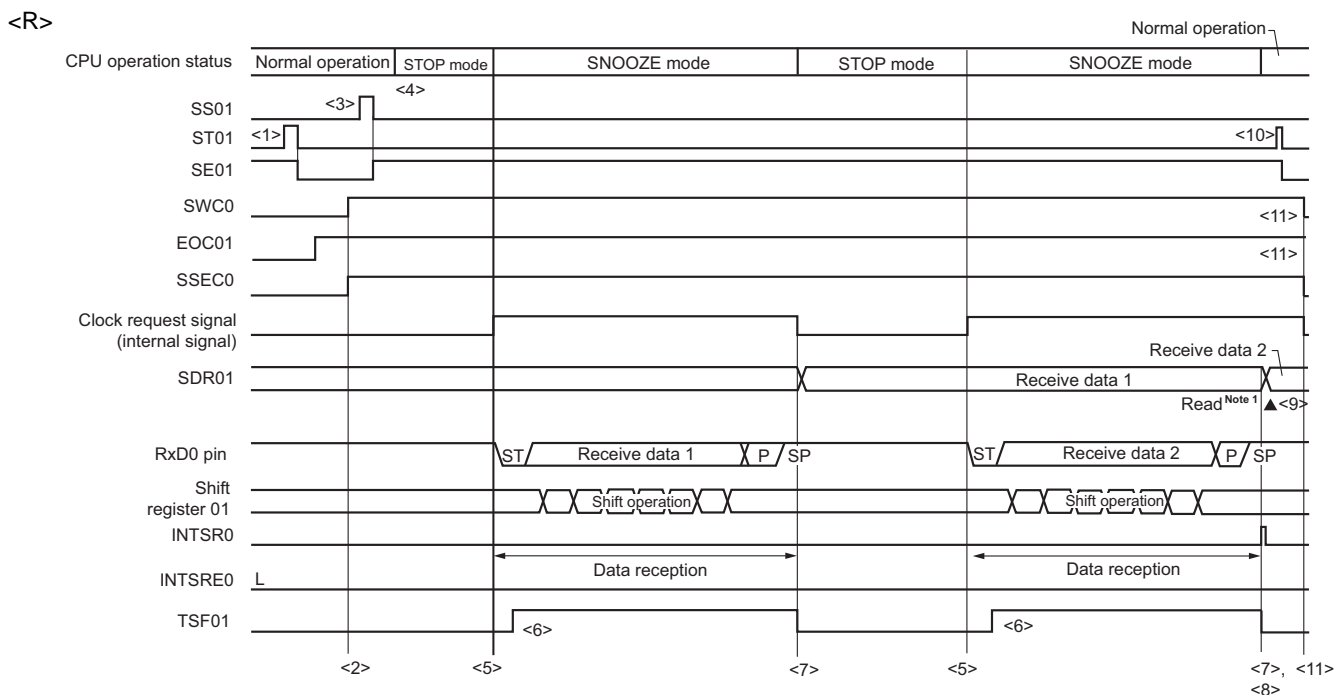
Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 14-90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECM = 0/1) and Figure 14-91 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECM = 0).

2. m = 0, 1; q = 0, 2

**(3) SNOOZE mode operation (EOCm1 = 1, SSECm = 1: Error interrupt (INTSREq) generation is stopped)**

Because EOCm1 = 1 and SSECm = 1, an error interrupt (INTSREq) is not generated when a communication error occurs.

**Figure 14-93. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**



**Note** Read the received data when SWCm = 1.

**Cautions 1.** Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEM1 bit, and stop the operation).

And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

**2.** If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFM1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFM1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (Rx/Dq register) (8 bits) or SDRm1[8:0] (9 bits).

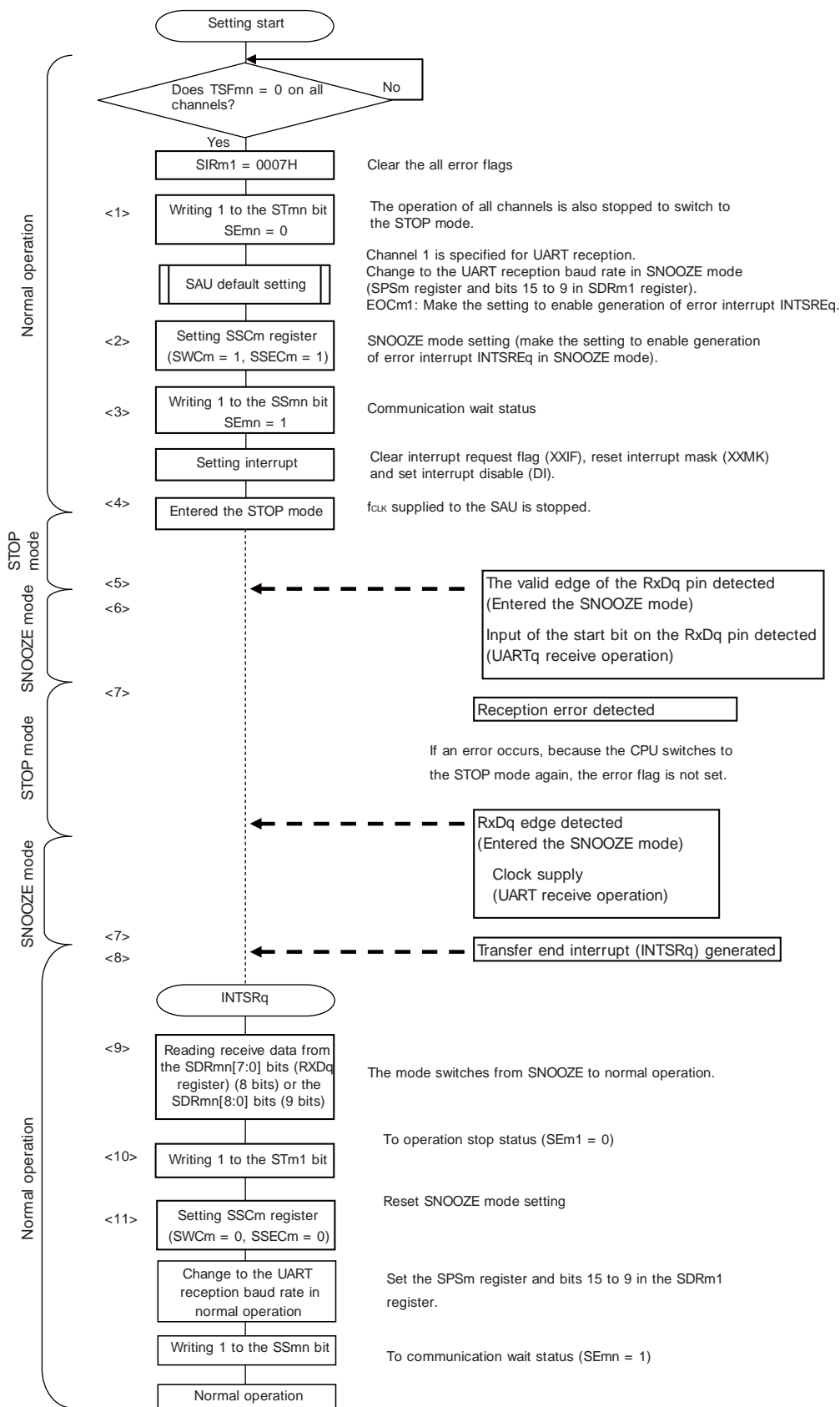
**Remarks 1.** <1> to <11> in the figure correspond to <1> to <11> in Figure 14-94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

**2.** m = 0, 1; q = 0, 2



Figure 14-94. Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECM = 1)

<R>



(Caution and Remarks are listed on the next page.)

**Caution** If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFM1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFM1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

- Remarks**
1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 14-93 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.
  2. m = 0, 1; q = 0, 2

#### 14.6.4 Calculating baud rate

##### (1) Baud rate calculation expression

The baud rate for UART (UART0 to UART3) communication can be calculated by the following expressions.

$$\text{(Baud rate)} = \{\text{Operation clock (f}_{\text{MCK}}\text{) frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2 \text{ [bps]}$$

**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, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

The operation clock (f<sub>MCK</sub>) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 14-4. Selection of Operation Clock For UART

SMRmn Register	SPSm Register								Operation Clock ( $f_{CLK}$ ) <sup>Note</sup>	
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	$f_{CLK} = 24 \text{ MHz}$
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	

**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) = 000FH) the operation of the serial array unit (SAU).

**Remarks 1.** X: Don't care

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

**(2) Baud rate error during transmission**

The baud rate error of UART (UART0 to UART3) 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.

$$(\text{Baud rate error}) = (\text{Calculated baud rate value}) \div (\text{Target baud rate}) \times 100 - 100 [\%]$$

Here is an example of setting a UART baud rate at  $f_{\text{CLK}} = 24 \text{ MHz}$ .

UART Baud Rate (Target Baud Rate)	$f_{\text{CLK}} = 24 \text{ MHz}$			
	Operation Clock ( $f_{\text{MCK}}$ )	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{\text{CLK}}/2^9$	77	300.48 bps	+0.16 %
600 bps	$f_{\text{CLK}}/2^8$	77	600.96 bps	+0.16 %
1200 bps	$f_{\text{CLK}}/2^7$	77	1201.92 bps	+0.16 %
2400 bps	$f_{\text{CLK}}/2^6$	77	2403.85 bps	+0.16 %
4800 bps	$f_{\text{CLK}}/2^5$	77	4807.69 bps	+0.16 %
9600 bps	$f_{\text{CLK}}/2^4$	77	9615.38 bps	+0.16 %
19200 bps	$f_{\text{CLK}}/2^3$	77	19230.8 bps	+0.16 %
31250 bps	$f_{\text{CLK}}/2^3$	47	31250.0 bps	$\pm 0.0$ %
38400 bps	$f_{\text{CLK}}/2^2$	77	38461.5 bps	+0.16 %
76800 bps	$f_{\text{CLK}}/2$	77	76923.1 bps	+0.16 %
153600 bps	$f_{\text{CLK}}$	77	153846 bps	+0.16 %
312500 bps	$f_{\text{CLK}}$	37	315789 bps	+1.05 %

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

**(3) Permissible baud rate range for reception**

The permissible baud rate range for reception during UART (UART0 to UART3) 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.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate}$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$$

Brate: Calculated baud rate value at the reception side (See 14.6.4 (1) Baud rate calculation expression.)

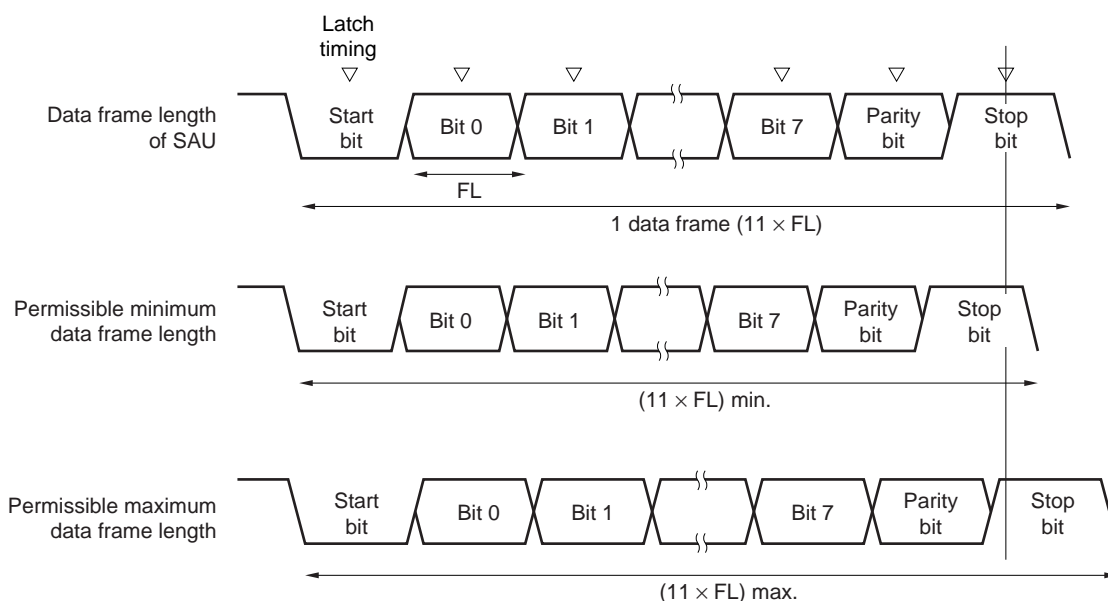
k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]

= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13

**Figure 14-95. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)**



As shown in Figure 14-95, 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.

**14.6.5 Procedure for processing errors that occurred during UART (UART0 to UART3) communication**

The procedure for processing errors that occurred during UART (UART0 to UART3) communication is described in Figures 14-96 and 14-97.

**Figure 14-96. 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 14-97. 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, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

## 14.7 LIN Communication Operation

### 14.7.1 LIN transmission

Of UART transmission, UART0 support LIN communication.

For LIN transmission, channel 0 of unit 0 is used.

UART	UART0	UART1	UART2	UART3
Support of LIN communication	Supported	Not supported	Not supported	Not supported
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	8 bits			
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR00[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] <sup>Note</sup>			
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)			
Parity bit	No parity bit			
Stop bit	Appending 1 bit			
Data direction	LSB first			

**Note** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = –40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = –40 to +105°C)**). In addition, LIN communication is usually 2.4/9.6/19.2 kbps is often used.

**Caution** 16-bit timer KB20 cannot be used together with the LIN-bus functions. When using 16-bit timer KB20, set the ISC register to its initial value (00H).

**Remark**  $f_{MCK}$ : Operation clock frequency of target channel  
 $f_{CLK}$ : System clock frequency



LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network.

Communication of LIN is single-master communication and up to 15 slaves can be connected to one master.

The slaves are used to control switches, actuators, and sensors, which are connected to the master via LIN.

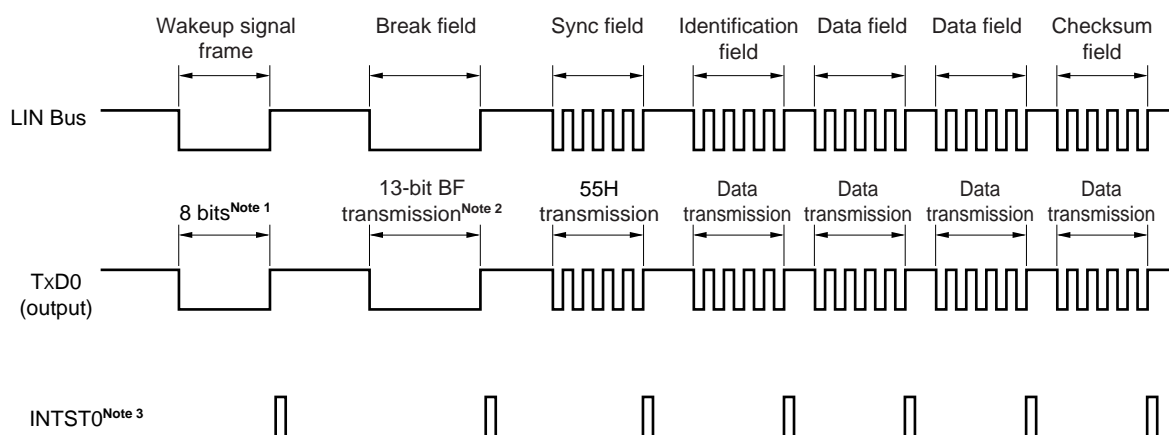
Usually, the master is connected to a network such as CAN (Controller Area Network).

A LIN bus is a single-wire bus to which nodes are connected via transceiver conforming to ISO9141.

According to the protocol of LIN, the master transmits a frame by attaching baud rate information to it. A slave receives this frame and corrects a baud rate error from the master. If the baud rate error of a slave is within ±15%, communication can be established.

Figure 14-98 outlines a master transmission operation of LIN.

Figure 14-98. Transmission Operation of LIN

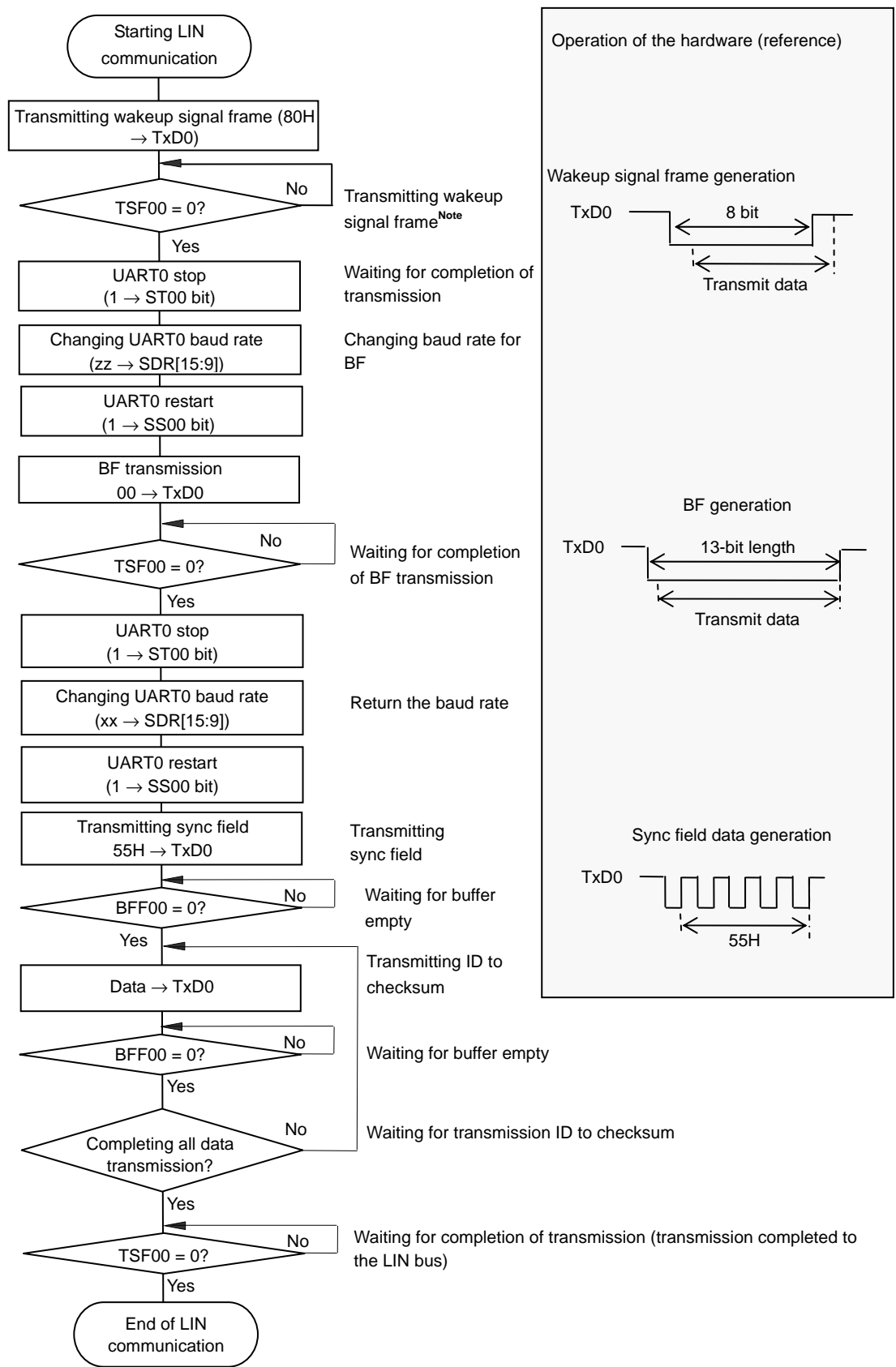


- Notes**
1. Set the baud rate in accordance with the wakeup signal regulations and transmit data of 80H.
  2. A break field is defined to have a width of 13 bits and output a low level. Where the baud rate for main transfer is N [bps], therefore, the baud rate of the break field is calculated as follows.  

$$\text{(Baud rate of break field)} = 9/13 \times N$$
 By transmitting data of 00H at this baud rate, a break field is generated.
  3. INTST0 is output upon completion of transmission. INTST0 is also output at BF transmission.

**Remark** The interval between fields is controlled by software.

Figure 14-99. Flowchart for LIN Transmission



**Note** When LIN-bus start from sleep status only

**Remark** Default setting of the UART is complete, and the flow from the transmission enable status.

### 14.7.2 LIN reception

Of UART reception, UART0 support LIN communication.

For LIN reception, channel 1 of unit 1 is used.

UART	UART0	UART1	UART2	UART3
Support of LIN communication	Supported	Not supported	Not supported	Not supported
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	<ul style="list-style-type: none"> <li>• Framing error detection flag (FEF01)</li> <li>• Overrun error detection flag (OVF01)</li> </ul>			
Transfer data length	8 bits			
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR01 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] <sup>Note</sup>			
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)			
Parity bit	No parity bit (The parity bit is not checked.)			
Stop bit	Check the first bit			
Data direction	LSB first			

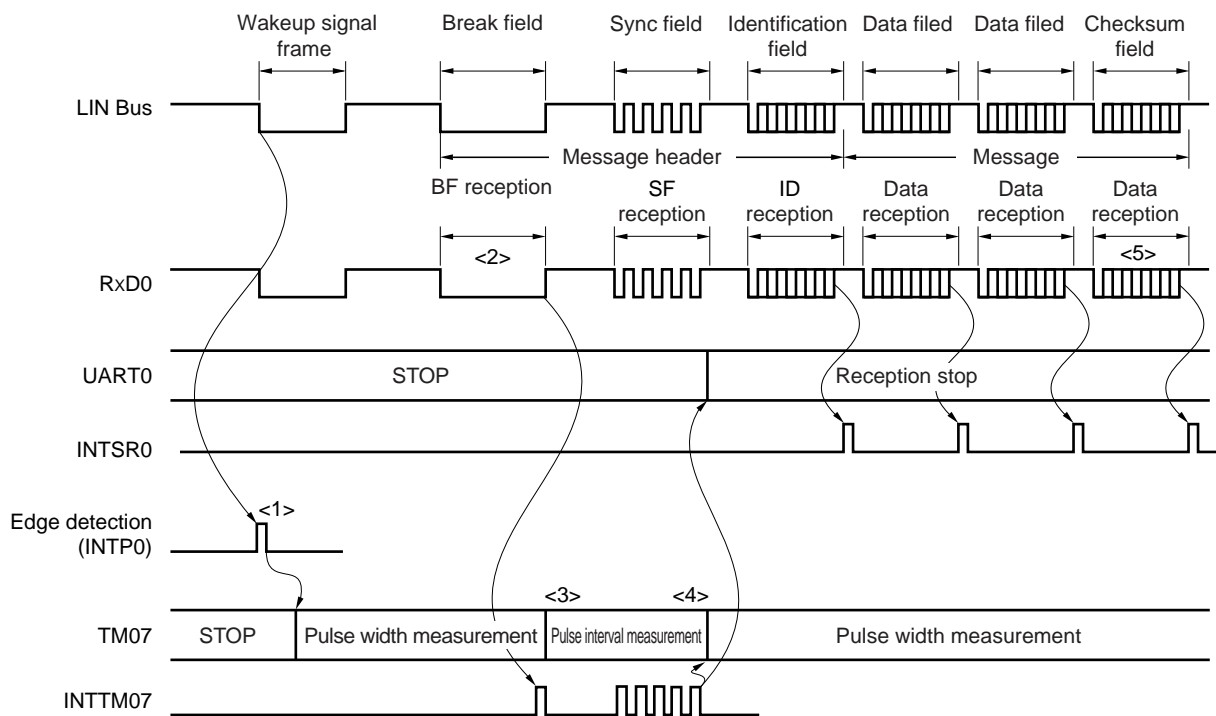
**Note** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = –40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = –40 to +105°C)**).

**Caution** 16-bit timer KB20 cannot be used together with the LIN-bus functions. When using 16-bit timer KB20, set the ISC register to its initial value (00H).

**Remark**  $f_{MCK}$ : Operation clock frequency of target channel  
 $f_{CLK}$ : System clock frequency

Figure 14-100 outlines a reception operation of LIN.

Figure 14-100. Reception Operation of LIN

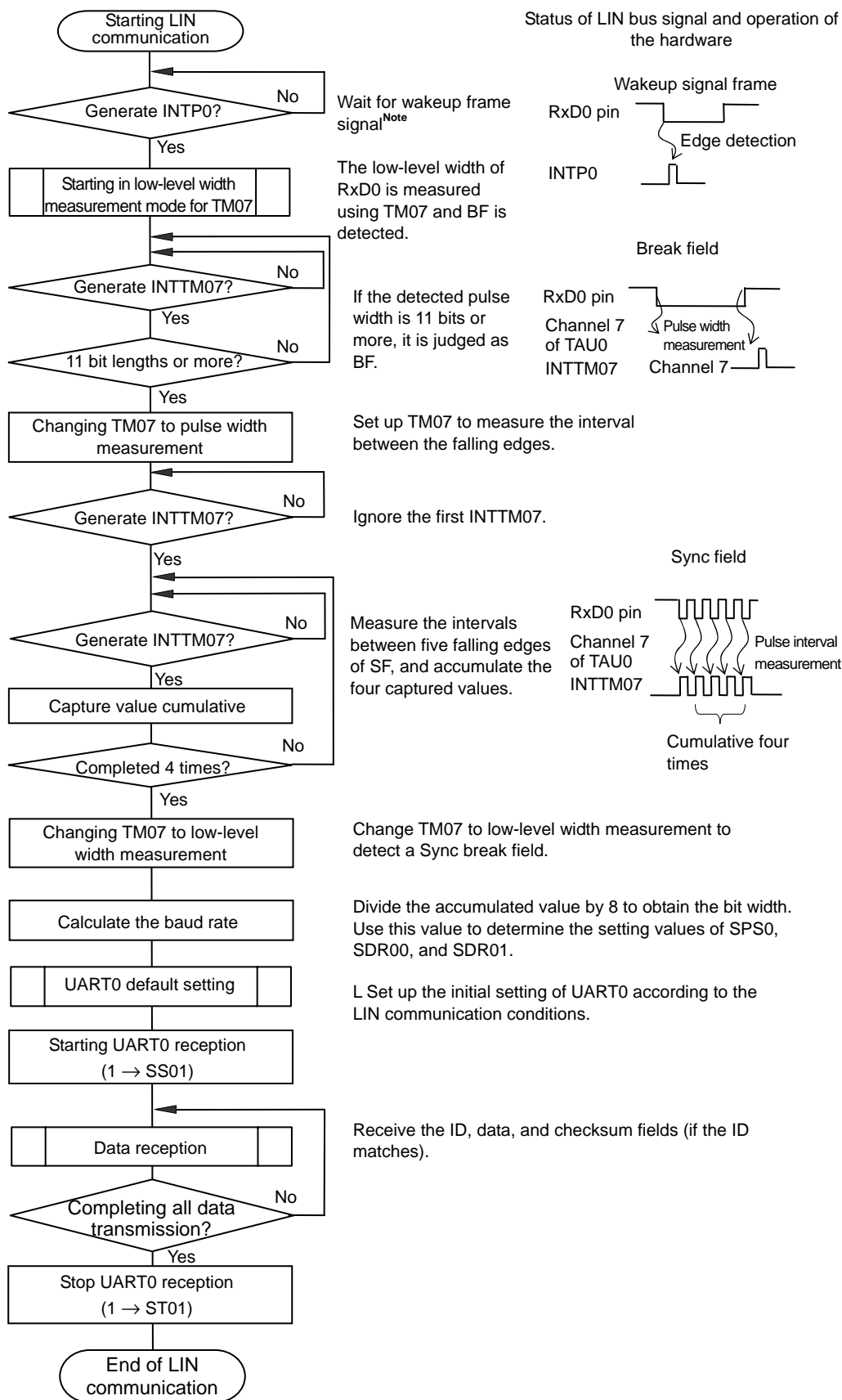


Here is the flow of signal processing.

- <1> The wakeup signal is detected by detecting an interrupt edge (INTP0) on a pin. When the wakeup signal is detected, change TM07 to pulse width measurement upon detection of the wakeup signal to measure the low-level width of the BF signal. Then wait for BF signal reception.
- <2> TM07 starts measuring the low-level width upon detection of the falling edge of the BF signal, and then captures the data upon detection of the rising edge of the BF signal. The captured data is used to judge whether it is the BF signal.
- <3> When the BF signal has been received normally, change TM07 to pulse interval measurement and measure the interval between the falling edges of the RxD0 signal in the Sync field four times (see **6.8.3 Operation as input pulse interval measurement**).
- <4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART2 once and adjust (re-set) the baud rate.
- <5> The checksum field should be distinguished by software. In addition, processing to initialize UART2 after the checksum field is received and to wait for reception of BF should also be performed by software.

Figure 14-101. Flowchart for LIN Reception

<R>



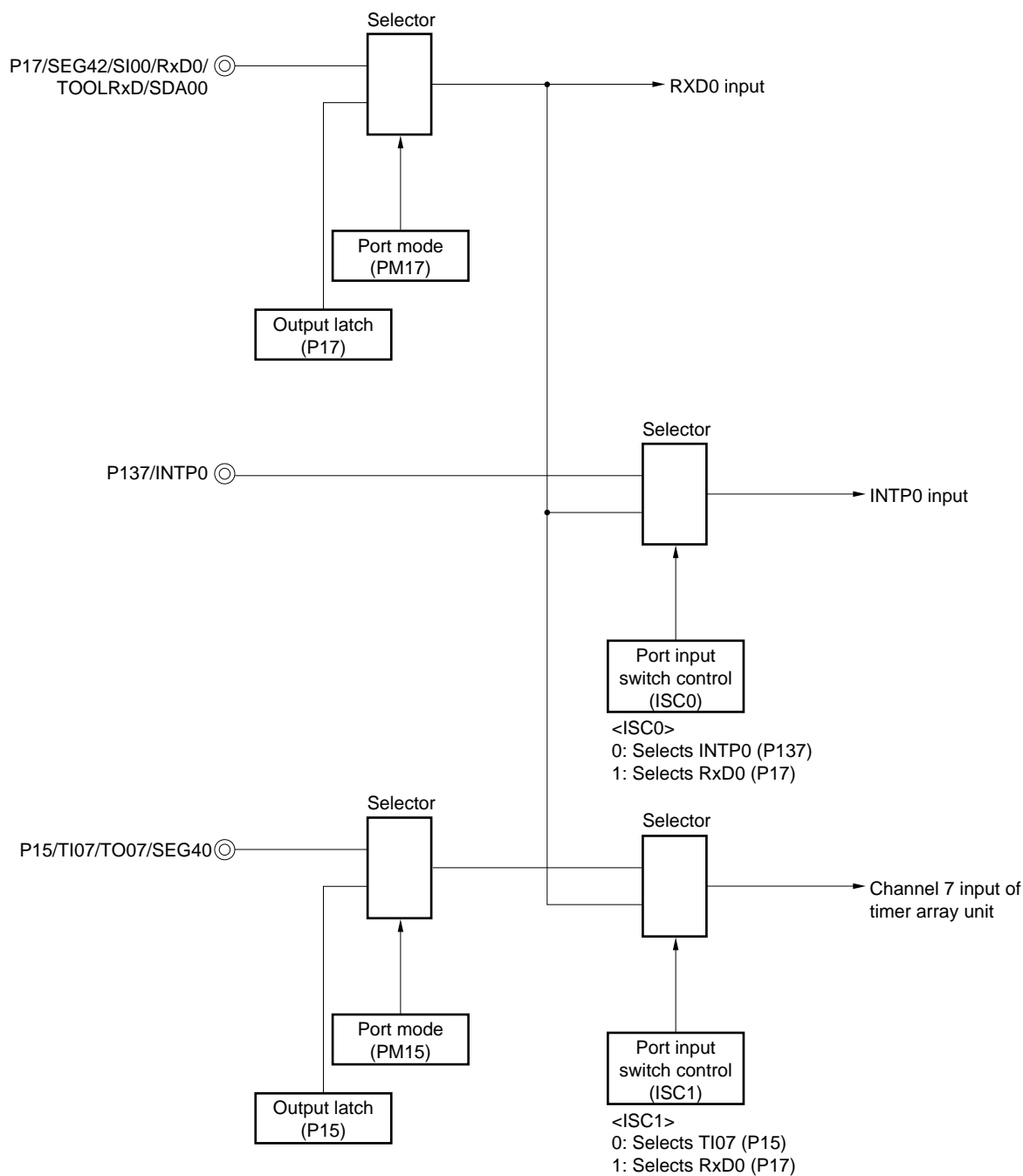
**Note** Required in the sleep status only.

Figure 14-102 shows the configuration of a port that manipulates reception of LIN.

The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (INTP0). The length of the sync field transmitted from the master can be measured by using the external event capture operation of the timer array unit 0 to calculate a baud-rate error.

By controlling switch of port input (ISC0/ISC1), the input source of port input (RxD0) for reception can be input to the external interrupt pin (INTP0) and timer array unit

**Figure 14-102. Port Configuration for Manipulating Reception of LIN**



**Remark** ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See Figure 14-21.)

The peripheral functions used for the LIN communication operation are as follows.

<Peripheral functions used>

- External interrupt (INTP0); Wakeup signal detection  
Usage: To detect an edge of the wakeup signal and the start of communication
- Channel 7 of timer array unit; Baud rate error detection, break field detection.  
Usage: To detect the length of the sync field (SF) and divide it by the number of bits in order to detect an error (The interval of the edge input to RxD0 is measured in the capture mode.)  
Measured the low-level width, determine whether break field (BF).
- Channels 0 and 1 (UART0) of serial array unit 0 (SAU0)

## 14.8 Operation of Simplified I<sup>2</sup>C (IIC00, IIC10) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master.

Operate the control registers by software for setting the start and stop conditions while observing the specifications of the I<sup>2</sup>C bus line

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function<sup>Note</sup> and ACK detection function
- Data length of 8 bits  
(When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Generation of start condition and stop condition for software

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Parity error (ACK error)

\* [Functions not supported by simplified I<sup>2</sup>C]

- Slave transmission, slave reception
- Multi-master function (arbitration loss detection function)
- Wait 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 **14.8.3 (2)** for details.

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02



The channel supporting simplified I<sup>2</sup>C (IIC00, IIC10) is channels 0 and 2 of SAU0.

- 64-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I <sup>2</sup> C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	UART2	–
	1	–		–

- 80-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I <sup>2</sup> C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	UART2	–
	1	–		–
	2	–	UART3	–
	3	–		–

Simplified I<sup>2</sup>C (IIC00, IIC10) performs the following four types of communication operations.

- Address field transmission (See 14.8.1.)
- Data transmission (See 14.8.2.)
- Data reception (See 14.8.3.)
- Stop condition generation (See 14.8.4.)

### 14.8.1 Address field transmission

Address field transmission is a transmission operation that first executes in I<sup>2</sup>C 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 I <sup>2</sup> C	IIC00	IIC10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCL00, SDA00 <sup>Note 1</sup>	SCL10, SDA10 <sup>Note 1</sup>
Interrupt	INTIIC00	INTIIC10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	ACK 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 <sup>Note 2</sup>	Max. $f_{MCK}/4$ [Hz] ( $SDR_{mn}[15:9] = 1$ or more) $f_{MCK}$ : Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I <sup>2</sup> C. <ul style="list-style-type: none"> <li>• Max. 1 MHz (fast mode plus)</li> <li>• Max. 400 kHz (fast mode)</li> <li>• Max. 100 kHz (standard mode)</li> </ul>	
Data level	Non-reversed 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<sup>2</sup>C, set the N-ch open-drain output ( $V_{DD}$  tolerance) mode (POM06, POM17 = 1) for the port output mode registers (POM0, POM1) (see **4.3 Registers Controlling Port Function** for details). When IIC00, IIC10 communicating with an external device with a different potential, set the N-ch open-drain output ( $V_{DD}$  tolerance) mode (POM05, POM16 = 1) also for the clock input/output pins (SCL00, SCL10) (see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers** for details).

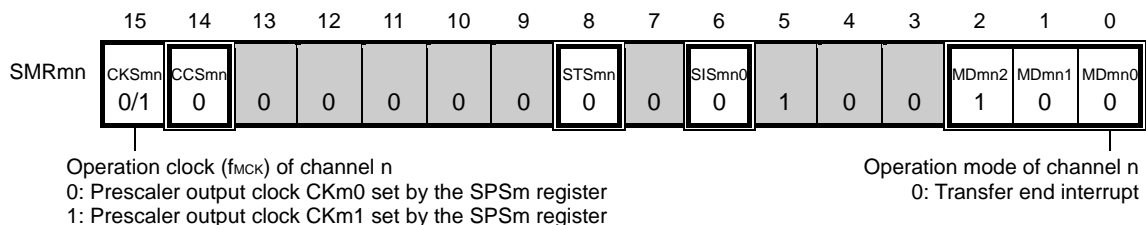
**2.** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)**).

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02

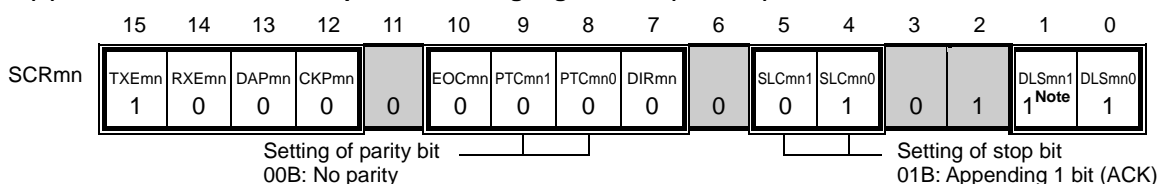
(1) Register setting

Figure 14-103. Example of Contents of Registers for Address Field Transmission of Simplified I<sup>2</sup>C (IIC00, IIC10) (1/2)

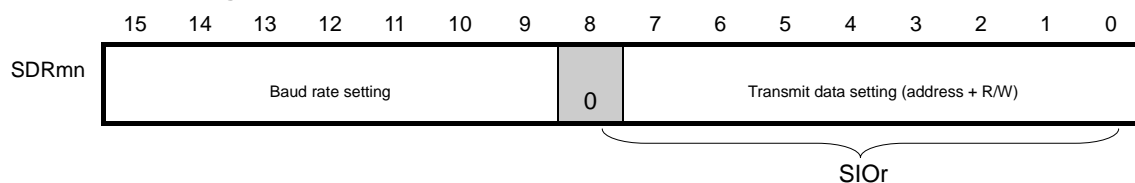
(a) Serial mode register mn (SMRmn)



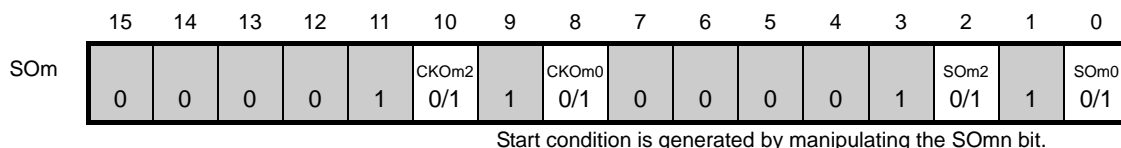
(b) Serial communication operation setting register mn (SCRmn)



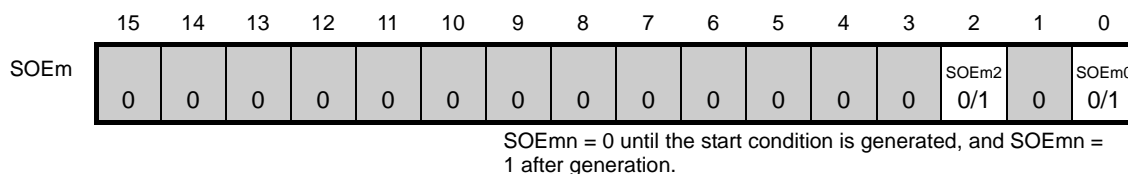
(c) Serial data register mn (SDRmn) (lower 8 bits: SIO<sub>r</sub>)



(d) Serial output register m (SOM<sub>m</sub>)



(e) Serial output enable register m (SOEm)



**Note** Only provided for the SCR00 register. This bit is fixed to 1 for the other registers.

- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10), mn = 00, 02
  - : 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 14-103. Example of Contents of Registers for Address Field Transmission of Simplified I<sup>2</sup>C (IIC00, IIC10) (2/2)**

(f) Serial channel start register m (SSm) ... Sets only the bits of the target channel is 1.

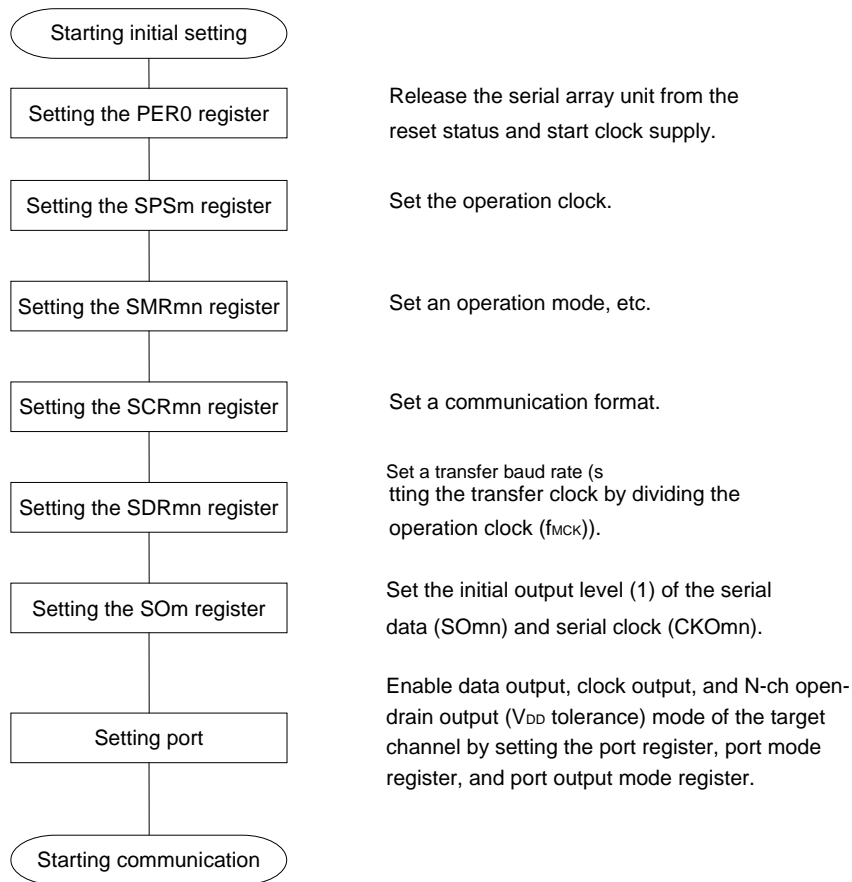
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

SSmn = 0 until the start condition is generated, and SSmn = 1 after generation.

- Remarks**
- m: Unit number (m = 0), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10), mn = 00, 02
  - : 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

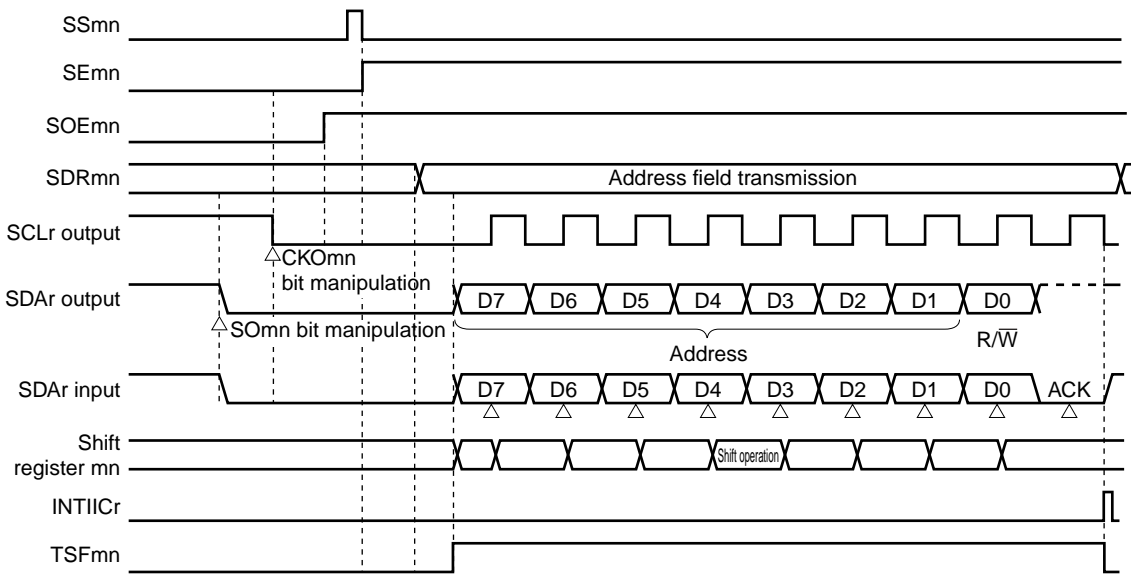
(2) Operation procedure

Figure 14-104. Initial Setting Procedure for Simplified I<sup>2</sup>C



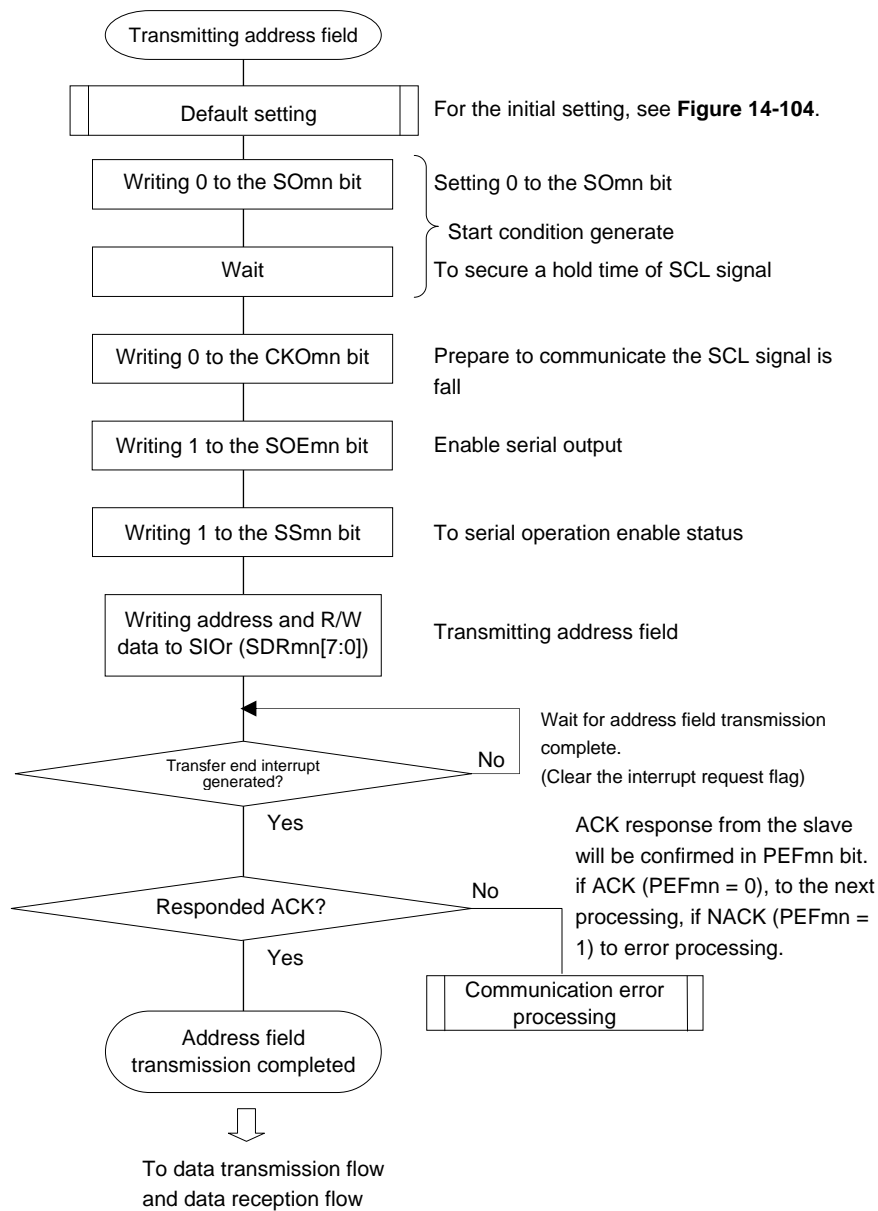
(3) Processing flow

Figure 14-105. Timing Chart of Address Field Transmission



**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10), mn = 00, 02

**Figure 14-106. Flowchart of Simplified I<sup>2</sup>C Address Field Transmission**



### 14.8.2 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 <sup>2</sup> C	IIC00	IIC10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCL00, SDA00 <sup>Note 1</sup>	SCL10, SDA10 <sup>Note 1</sup>
Interrupt	INTIIC00	INTIIC10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	ACK error flag (PEFmn)	
Transfer data length	8 bits	
Transfer rate <sup>Note 2</sup>	Max. $f_{MCK}/4$ [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 <sup>2</sup> C. <ul style="list-style-type: none"> <li>• Max. 1 MHz (fast mode plus)</li> <li>• Max. 400 kHz (fast mode)</li> <li>• Max. 100 kHz (standard mode)</li> </ul>	
Data level	Non-reversed 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<sup>2</sup>C, set the N-ch open-drain output ( $V_{DD}$  tolerance) mode (POM06, POM17 = 1) for the port output mode registers (POM0, POM1) (see **4.3 Registers Controlling Port Function** for details). When IIC00, IIC10 communicating with an external device with a different potential, set the N-ch open-drain output ( $V_{DD}$  tolerance) mode (POM05, POM16 = 1) also for the clock input/output pins (SCL00, SCL10) (see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers** for details).

**2.** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)**).

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02



(1) Register setting

Figure 14-107. Example of Contents of Registers for Data Transmission of Simplified I<sup>2</sup>C (IIC00, IIC10) (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	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	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	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIO<sub>r</sub>) ... During data transmission/reception, valid only lower 8-bits (SIO<sub>r</sub>)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting <sup>Note 2</sup>							0	Transmit data setting							
	SIO <sub>r</sub>															

(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						CKOm2		CKOm0						SOM2		SOM0
	0	0	0	0	1	0/1 <sup>Note 3</sup>	1	0/1 <sup>Note 3</sup>	0	0	0	0	1	0/1 <sup>Note 3</sup>	1	0/1 <sup>Note 3</sup>

(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														SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

- Notes**
1. Only provided for the SCR00 register. This bit is fixed to 1 for the other registers.
  2. Because the setting is completed by address field transmission, setting is not required.
  3. The value varies depending on the communication data during communication operation.

- Remarks**
1. m: Unit number (m = 0), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10), mn = 00, 02
  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 14-107. Example of Contents of Registers for Data Transmission of Simplified I<sup>2</sup>C (IIC00, IIC10) (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	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

**Remarks 1.** m: Unit number (m = 0), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10), mn = 00, 02

**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 14-108. Timing Chart of Data Transmission

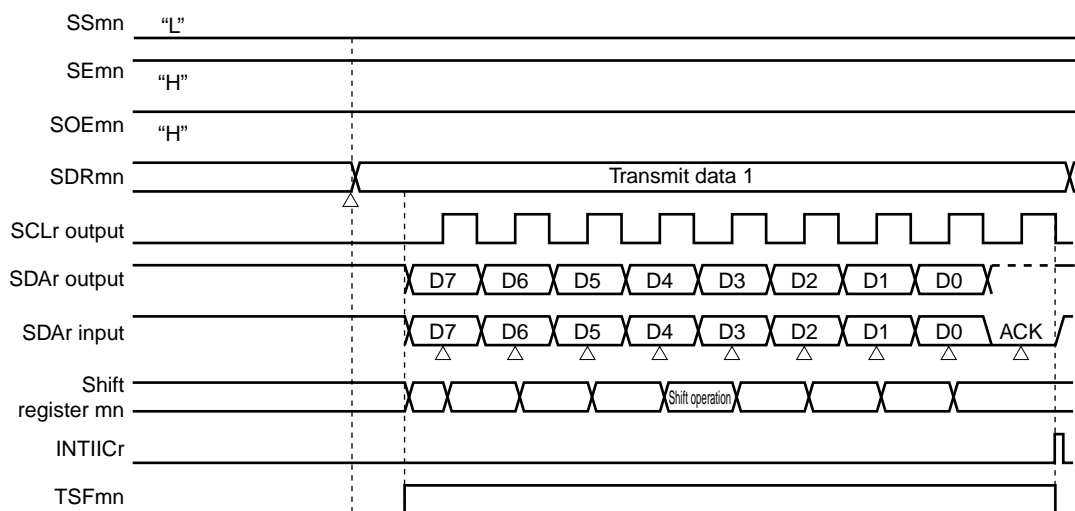
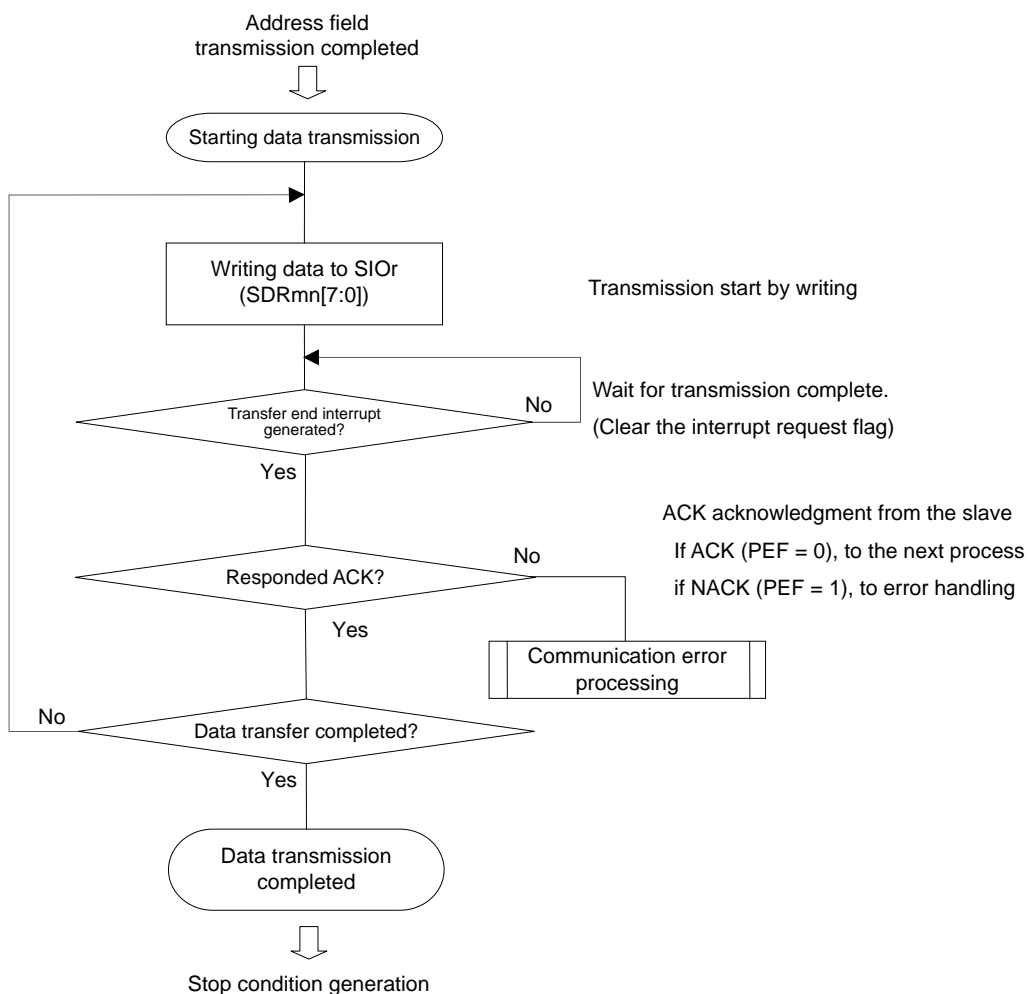


Figure 14-109. Flowchart of Simplified I<sup>2</sup>C Data Transmission



### 14.8.3 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 <sup>2</sup> C	IIC00	IIC10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCL00, SDA00 <sup>Note 1</sup>	SCL10, SDA10 <sup>Note 1</sup>
Interrupt	INTIIC00	INTIIC10
	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 <sup>Note 2</sup>	Max. $f_{MCK}/4$ [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 <sup>2</sup> C. <ul style="list-style-type: none"> <li>• Max. 1 MHz (fast mode plus)</li> <li>• Max. 400 kHz (fast mode)</li> <li>• Max. 100 kHz (standard mode)</li> </ul>	
Data level	Non-reversed 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<sup>2</sup>C, set the N-ch open-drain output ( $V_{DD}$  tolerance) mode (POM06, POM17 = 1) for the port output mode registers (POM0, POM1) (see **4.3 Registers Controlling Port Function** for details). When IIC00, IIC10 communicating with an external device with a different potential, set the N-ch open-drain output ( $V_{DD}$  tolerance) mode (POM05, POM16 = 1) also for the clock input/output pins (SCL00, SCL10) (see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers** for details).

**2.** Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)**).

**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02

(1) Register setting

Figure 14-110. Example of Contents of Registers for Data Reception of Simplified I<sup>2</sup>C (IIC00, IIC10) (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	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	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	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIO<sub>r</sub>)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting <sup>Note 2</sup>								0	Dummy transmit data setting (FFH)						
	SIO <sub>r</sub>															

(d) Serial output register m (SOM<sub>m</sub>) ... 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 <sub>m</sub>						CKOm2		CKOm0						SOM2		SOM0
	0	0	0	0	1	0/1 <sup>Note 3</sup>	1	0/1 <sup>Note 3</sup>	0	0	0	0	1	0/1 <sup>Note 3</sup>	1	0/1 <sup>Note 3</sup>

(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														SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0	0/1

- Notes**
1. Only provided for the SCR00 register. This bit is fixed to 1 for the other registers.
  2. The baud rate setting is not required because the baud rate has already been set when the address field was transmitted.
  3. The value varies depending on the communication data during communication operation.

- Remarks**
1. m: Unit number (m = 0), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10), mn = 00, 02
  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 14-110. Example of Contents of Registers for Data Reception of Simplified I<sup>2</sup>C (IIC00, IIC10) (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	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

**Remarks 1.** m: Unit number (m = 0), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10), mn = 00, 02

**2.** : Setting is fixed in the CSI master transmission mode, : Setting disabled (set to the initial value)

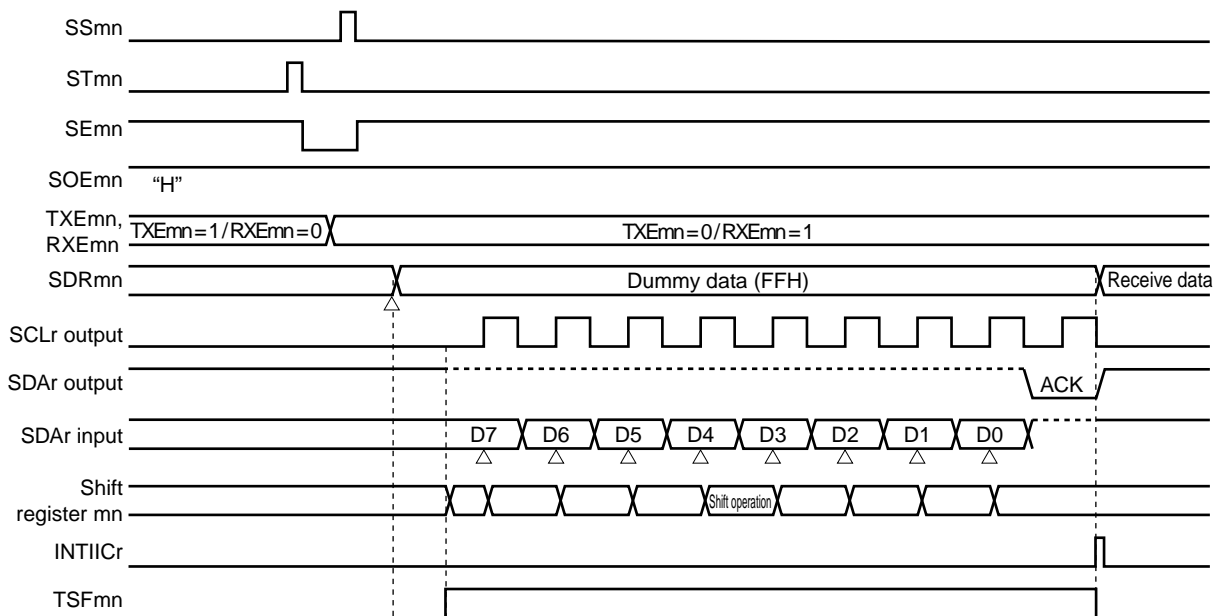
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

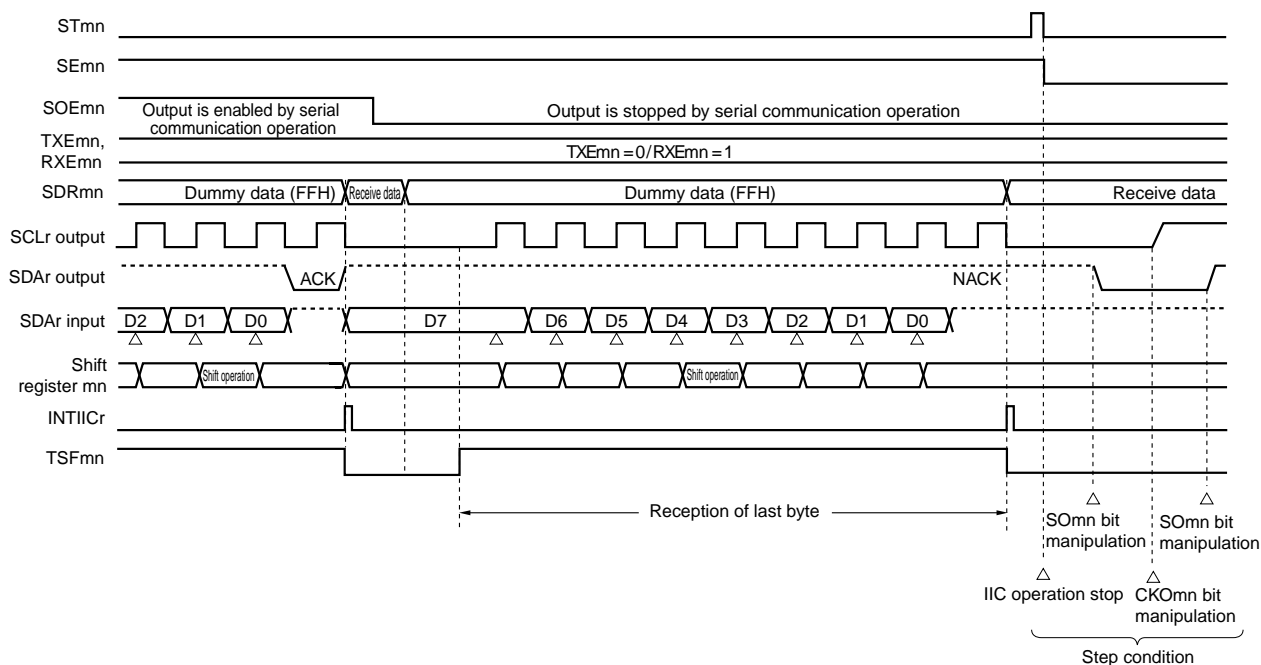
(2) Processing flow

Figure 14-111. Timing Chart of Data Reception

(a) When starting data reception

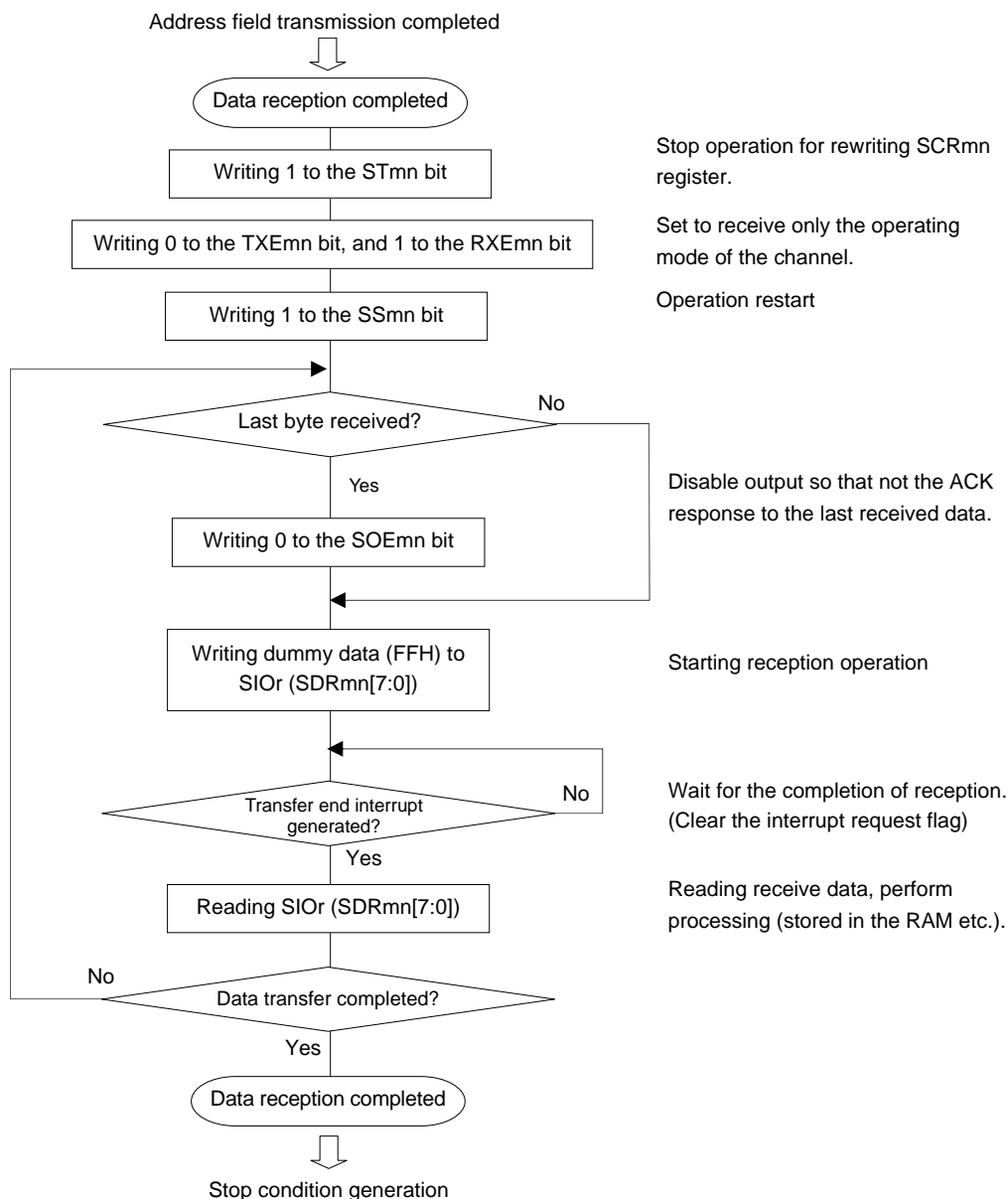


(b) When receiving last data



**Remark** m: Unit number (m = 0), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10), mn = 00, 02

**Figure 14-112. 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.

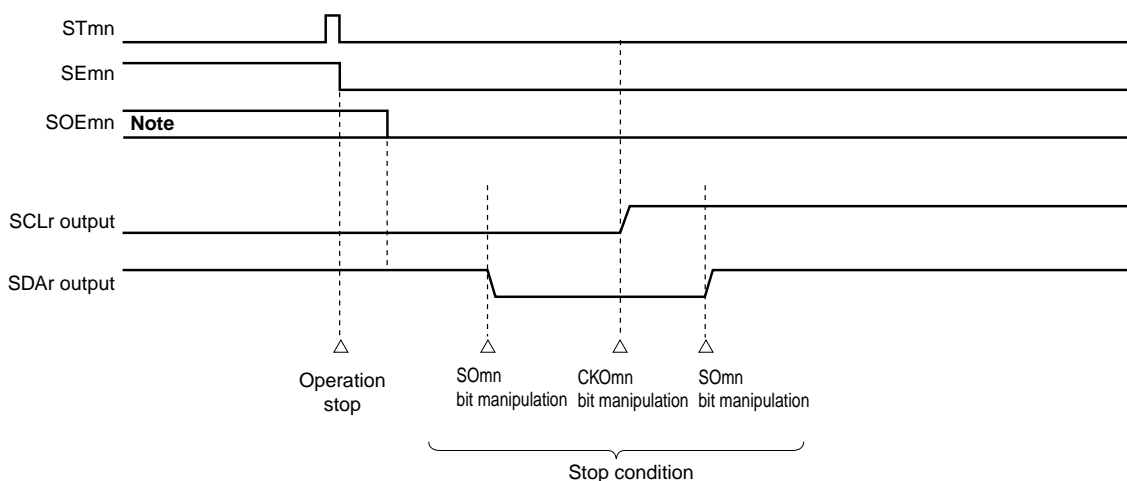


14.8.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

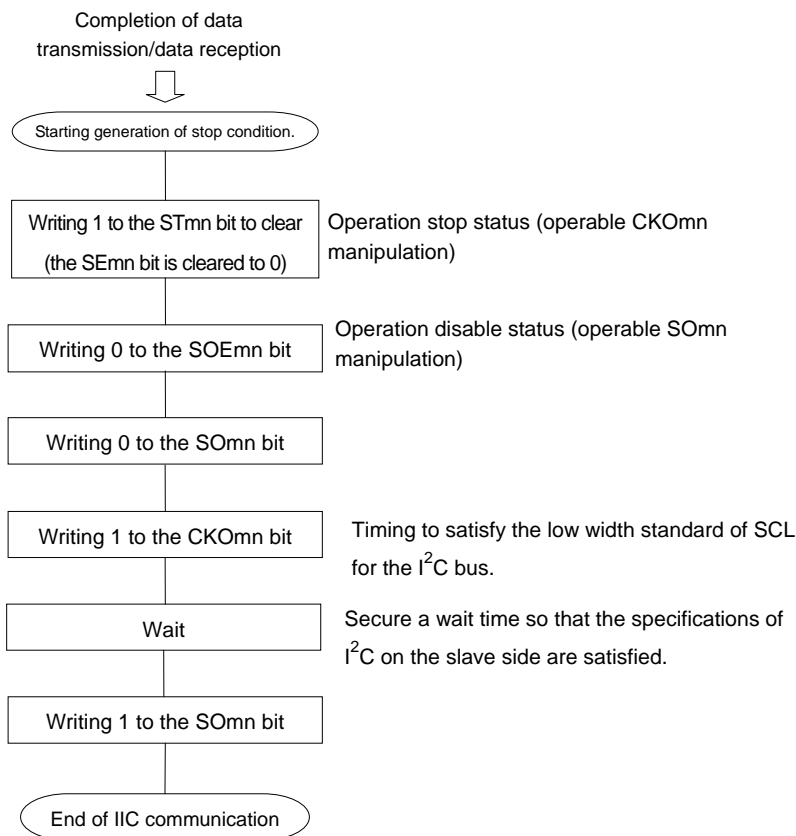
(1) Processing flow

Figure 14-113. 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 14-114. Flowchart of Stop Condition Generation



### 14.8.5 Calculating transfer rate

The transfer rate for simplified I<sup>2</sup>C (IIC00, IIC10) communication can be calculated by the following expressions.

$$\text{(Transfer rate)} = \{\text{Operation clock (f}_{\text{MCK}}\text{) frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2$$

**Caution** SDRmn[15:9] must not be set to 0000000B. Be sure to set a value of 0000001B or greater for SDRmn[15:9]. The duty ratio of the SCL signal output by the simplified I2C is 50%. The I2C bus specifications define that the low-level width of the SCL signal is longer than the high-level width. If 400 kbps (fast mode) or 1 Mbps (fast mode plus) is specified, therefore, the low-level width of the SCL output signal becomes shorter than the value specified in the I2C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I2C bus specifications.

- 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, 2), mn = 00, 02

The operation clock (f<sub>MCK</sub>) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

**Table 14-5. Selection of Operation Clock For Simplified I<sup>2</sup>C**

SMRmn Register	SPSm Register								Operation Clock (f <sub>MCK</sub> ) <sup>Note</sup>	
	CKSmn	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00	f <sub>CLK</sub> = 24 MHz
0	X	X	X	X	0	0	0	0	f <sub>CLK</sub>	24 MHz
	X	X	X	X	0	0	0	1	f <sub>CLK</sub> /2	12 MHz
	X	X	X	X	0	0	1	0	f <sub>CLK</sub> /2 <sup>2</sup>	6 MHz
	X	X	X	X	0	0	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	3 MHz
	X	X	X	X	0	1	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	1.5 MHz
	X	X	X	X	0	1	0	1	f <sub>CLK</sub> /2 <sup>5</sup>	750 kHz
	X	X	X	X	0	1	1	0	f <sub>CLK</sub> /2 <sup>6</sup>	375 kHz
	X	X	X	X	0	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	187.5 kHz
	X	X	X	X	1	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	93.8 kHz
	X	X	X	X	1	0	0	1	f <sub>CLK</sub> /2 <sup>9</sup>	46.9 kHz
	X	X	X	X	1	0	1	0	f <sub>CLK</sub> /2 <sup>10</sup>	23.4 kHz
X	X	X	X	1	0	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	11.7 kHz	
1	0	0	0	0	X	X	X	X	f <sub>CLK</sub>	24 MHz
	0	0	0	1	X	X	X	X	f <sub>CLK</sub> /2	12 MHz
	0	0	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>2</sup>	6 MHz
	0	0	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>3</sup>	3 MHz
	0	1	0	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>4</sup>	1.5 MHz
	0	1	0	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>5</sup>	750 kHz
	0	1	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>6</sup>	375 kHz
	0	1	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>7</sup>	187.5 kHz
	1	0	0	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>8</sup>	93.8 kHz
	1	0	0	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>9</sup>	46.9 kHz
	1	0	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>10</sup>	23.4 kHz
1	0	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>11</sup>	11.7 kHz	
Other than above									Setting prohibited	

**Note** When changing the clock selected for f<sub>CLK</sub> (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) 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, 2), mn = 00, 02

Here is an example of setting an I<sup>2</sup>C transfer rate where f<sub>MCK</sub> = f<sub>CLK</sub> = 24 MHz.

I <sup>2</sup> C Transfer Mode (Desired Transfer Rate)	f <sub>CLK</sub> = 24 MHz			
	Operation Clock (f <sub>MCK</sub> )	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	f <sub>CLK</sub> /2	59	100 kHz	0.0%
400 kHz	f <sub>CLK</sub>	29	380 kHz	5.0% <sup>Note</sup>
1 MHz	f <sub>CLK</sub>	5	0.84 MHz	16.0% <sup>Note</sup>

**Note** The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

**14.8.6 Procedure for processing errors that occurred during simplified I<sup>2</sup>C (IIC00, IIC10) communication**

The procedure for processing errors that occurred during simplified I<sup>2</sup>C (IIC00, IIC10) communication is described in Figures 14-115 and 14-116.

**Figure 14-115. 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 14-116. Processing Procedure in Case of Parity Error (ACK error) in Simplified I<sup>2</sup>C Mode**

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 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, 2), r: IIC number (r = 00, 10), mn = 00, 02

## CHAPTER 15 SERIAL INTERFACE IICA

### 15.1 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<sup>2</sup>C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLAn) line and a serial data bus (SDAAn) line.

This mode complies with the I<sup>2</sup>C bus format and the master device can generate “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<sup>2</sup>C bus.

Since the SCLAn and SDAAn 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 (INTIICAn) 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 WUPn bit of IICA control register n1 (IICCTLn1).

Figure 15-1 shows a block diagram of serial interface IICA.

**Remark** n = 0

Figure 15-1. Block Diagram of Serial Interface IICA0

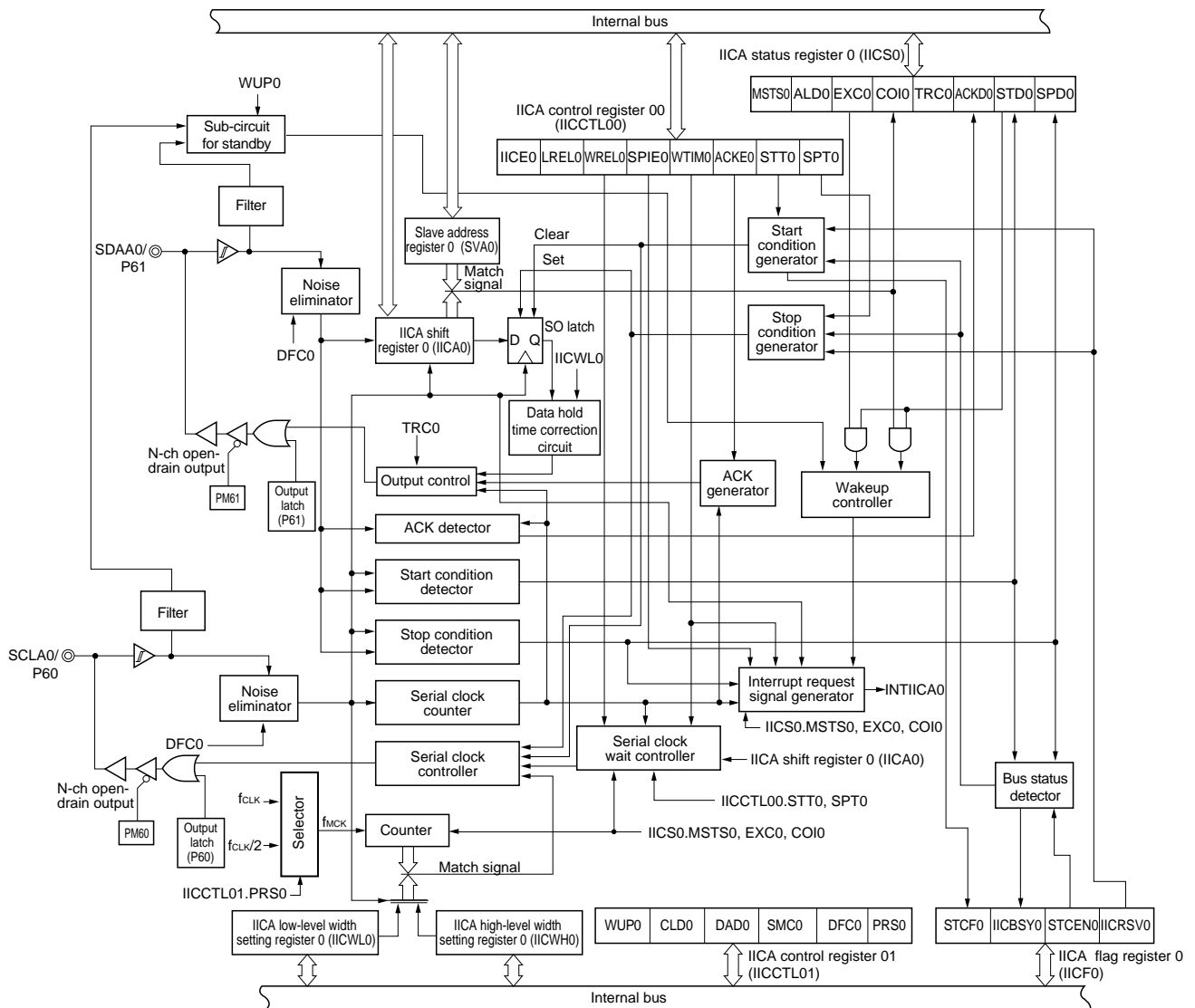
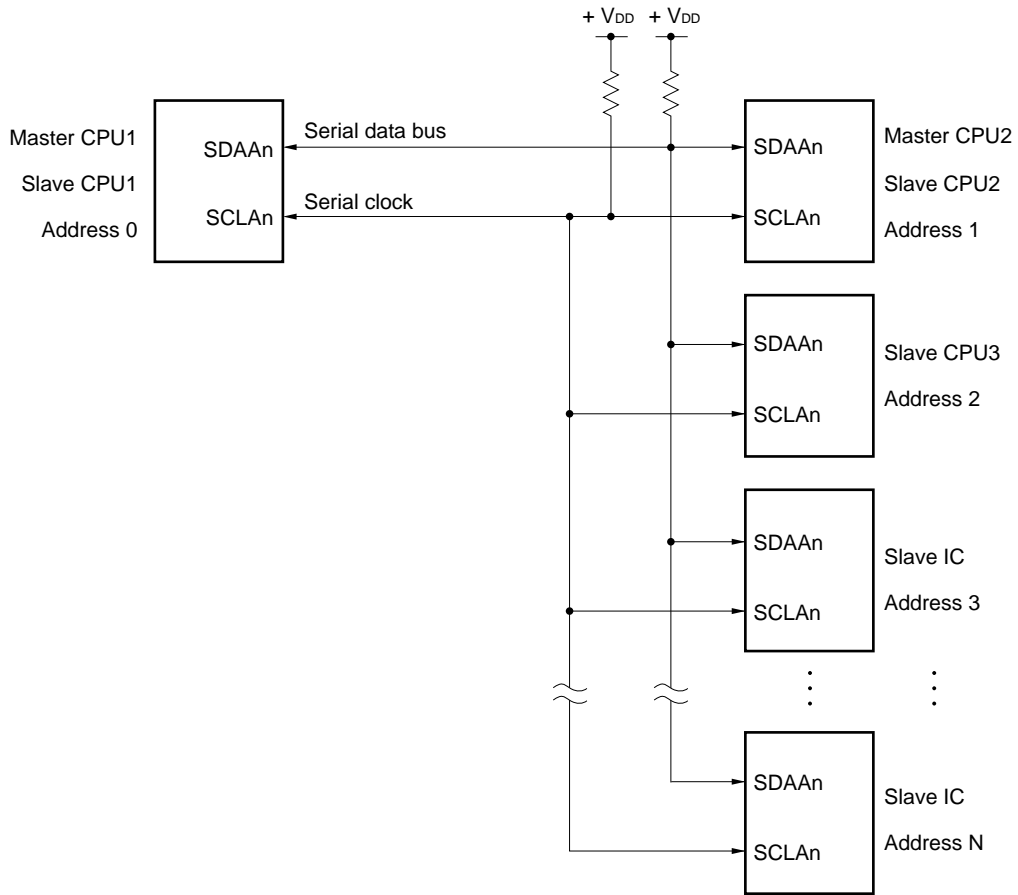


Figure 15-2 shows a serial bus configuration example.

Figure 15-2. Serial Bus Configuration Example Using I<sup>2</sup>C Bus



**Remark** n = 0

### 15.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

**Table 15-1. Configuration of Serial Interface IICA**

Item	Configuration
Registers	IICA shift register n (IICAn) Slave address register n (SVAn)
Control registers	Peripheral enable register 0 (PER0) IICA control register n0 (IICCTLn0) IICA status register n (IICSn) IICA flag register n (IICFn) IICA control register n1 (IICCTLn1) IICA low-level width setting register n (IICWLn) IICA high-level width setting register n (IICWHn) Port mode register 6 (PM6) Port register 6 (P6)

**Remark** n = 0

#### (1) IICA shift register n (IICAn)

The IICAn register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICAn 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 IICAn register.

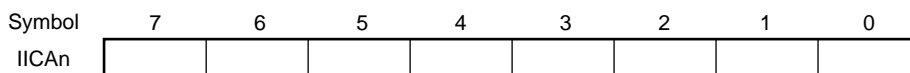
Cancel the wait state and start data transfer by writing data to the IICAn register during the wait period.

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

Reset signal generation clears IICAn to 00H.

**Figure 15-3. Format of IICA Shift Register n (IICAn)**

Address: FFF50H After reset: 00H R/W



- Cautions**
1. Do not write data to the IICAn register during data transfer.
  2. Write or read the IICAn register only during the wait period. Accessing the IICAn register in a communication state other than during the wait period is prohibited. When the device serves as the master, however, the IICAn register can be written only once after the communication trigger bit (STTn) is set to 1.
  3. When communication is reserved, write data to the IICAn register after the interrupt triggered by a stop condition is detected.

**Remark** n = 0



**(2) Slave address register n (SVAn)**

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode. The SVAn register can be set by an 8-bit memory manipulation instruction. However, rewriting to this register is prohibited while STDn = 1 (while the start condition is detected). Reset signal generation clears the SVAn register to 00H.

**Figure 15-4. Format of Slave Address Register n (SVAn)**

Address: F0234H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
SVAn	A6	A5	A4	A3	A2	A1	A0	0 <sup>Note</sup>

**Note** Bit 0 is fixed to 0.

**(3) SO latch**

The SO latch is used to retain the SDAAn pin's output level.

**(4) Wakeup controller**

This circuit generates an interrupt request (INTIICAn) when the address received by this register matches the address value set to the slave address register n (SVAn) 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 (INTIICAn).

An I<sup>2</sup>C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by the WTIMn bit)
- Interrupt request generated when a stop condition is detected (set by the SPIEn bit)

**Remark** WTIMn bit: Bit 3 of IICA control register n0 (IICCTLn0)

SPIEn bit: Bit 4 of IICA control register n0 (IICCTLn0)

**(7) Serial clock controller**

In master mode, this circuit generates the clock output via the SCLAn pin from a sampling clock.

**(8) Serial clock wait controller**

This circuit controls the wait 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.

**Remark** n = 0

**(11) Start condition generator**

This circuit generates a start condition when the STTn bit is set to 1.

However, in the communication reservation disabled status (IICRSVn bit = 1), when the bus is not released (IICBSYn bit = 1), start condition requests are ignored and the STCFn bit is set to 1.

**(12) Stop condition generator**

This circuit generates a stop condition when the SPTn 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 STCENn bit.

- Remarks 1.**
- |              |  |
|--------------|--|
| STTn bit:    | Bit 1 of IICA control register n0 (IICCTLn0) |
| SPTn bit:    | Bit 0 of IICA control register n0 (IICCTLn0) |
| IICRSVn bit: | Bit 0 of IICA flag register n (IICFn)        |
| IICBSYn bit: | Bit 6 of IICA flag register n (IICFn)        |
| STCFn bit:   | Bit 7 of IICA flag register n (IICFn)        |
| STCENn bit:  | Bit 1 of IICA flag register n (IICFn)        |
- 2.** n = 0

### 15.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following eight registers.

- Peripheral enable register 0 (PER0)
- IICA control register n0 (IICCTLn0)
- IICA flag register n (IICFn)
- IICA status register n (IICSn)
- IICA control register n1 (IICCTLn1)
- IICA low-level width setting register n (IICWLn)
- IICA high-level width setting register n (IICWHn)
- Port mode register 6 (PM6)
- Port register 6 (P6)

**Remark** n = 0

#### 15.3.1 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 IICAn 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 15-5. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

IICAnEN	Control of serial interface IICAn input clock supply
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by serial interface IICAn cannot be written.</li> <li>• Serial interface IICAn is in the reset status.</li> </ul>
1	Enables input clock supply. <ul style="list-style-type: none"> <li>• SFR used by serial interface IICAn can be read/written.</li> </ul>

**Cautions** 1. When setting serial interface IICA, be sure to set the following registers first while the IICAnEN bit is set to 1. If IICAnEN = 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 n0 (IICCTLn0)
- IICA flag register n (IICFn)
- IICA status register n (IICSn)
- IICA control register n1 (IICCTLn1)
- IICA low-level width setting register n (IICWLn)
- IICA high-level width setting register n (IICWHn)

2. Be sure to clear bits 1, 6 to 0.

**Remark** n = 0

### 15.3.2 IICA control register n0 (IICCTLn0)

This register is used to enable/stop I<sup>2</sup>C operations, set wait timing, and set other I<sup>2</sup>C operations.

The IICCTLn0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIEn, WTIMn, and ACKEn bits while IICEn = 0 or during the wait period. These bits can be set at the same time when the IICEn bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

**Remark** n = 0

**Figure 15-6. Format of IICA Control Register n0 (IICCTLn0) (1/4)**

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I <sup>2</sup> C operation enable
0	Stop operation. Reset the IICA status register n (IICSn) <sup>Note 1</sup> . Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCLAn and SDAAn lines are at high level.	
Condition for clearing (IICEn = 0)	
<ul style="list-style-type: none"> <li>• Cleared by instruction</li> <li>• Reset</li> </ul>	Condition for setting (IICEn = 1)
<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>	

LRELn <sup>Notes 2,3</sup>	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 SCLAn and SDAAn lines are set to high impedance. The following flags of IICA control register n0 (IICCTLn0) and the IICA status register n (IICSn) are cleared to 0. • STTn • SPTn • MSTSn • EXCn • COIn • TRCn • ACKDn • STDn
The standby mode following exit from communications remains in effect until the following communications entry conditions are met.	
<ul style="list-style-type: none"> <li>• After a stop condition is detected, restart is in master mode.</li> <li>• An address match or extension code reception occurs after the start condition.</li> </ul>	
Condition for clearing (LRELn = 0)	
<ul style="list-style-type: none"> <li>• Automatically cleared after execution</li> <li>• Reset</li> </ul>	Condition for setting (LRELn = 1)
<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>	

WRELn <sup>Notes 2,3</sup>	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When the WRELn bit is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRCn = 1), the SDAAn line goes into the high impedance state (TRCn = 0).	
Condition for clearing (WRELn = 0)	
<ul style="list-style-type: none"> <li>• Automatically cleared after execution</li> <li>• Reset</li> </ul>	Condition for setting (WRELn = 1)
<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>	

- Notes**
1. The IICA status register n (IICSn), the STCFn and IICBSYn bits of the IICA flag register n (IICFn), and the CLDn and DADn bits of IICA control register n1 (IICCTLn1) are reset.
  2. The signal of this bit is invalid while IICEn is 0.
  3. When the LRELn and WRELn bits are read, 0 is always read.

**Caution** If the operation of I<sup>2</sup>C is enabled (IICEn = 1) when the SCLAn line is high level, the SDAAn line is low level, and the digital filter is turned on (DFCn bit of IICCTLn1 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LRELn bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I<sup>2</sup>C (IICEn = 1).

**Remark** n = 0

Figure 15-6. Format of IICA Control Register n0 (IICCTLn0) (2/4)

SPIEn <sup>Note 1</sup>	Enable/disable generation of interrupt request when stop condition is detected	
0	Disable	
1	Enable	
If the WUPn bit of IICA control register n1 (IICCTLn1) is 1, no stop condition interrupt will be generated even if SPIEn = 1.		
Condition for clearing (SPIEn = 0)		Condition for setting (SPIEn = 1)
<ul style="list-style-type: none"> <li>• Cleared by instruction</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>

WTIMn <sup>Note 1</sup>	Control of wait 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 wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait 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 wait is set. Slave mode: After input of nine clocks, the clock is set to low level and wait 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 wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after an acknowledge (ACK) is issued. However, when the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.		
Condition for clearing (WTIMn = 0)		Condition for setting (WTIMn = 1)
<ul style="list-style-type: none"> <li>• Cleared by instruction</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>

ACKEn <sup>Notes 1, 2</sup>	Acknowledgment control	
0	Disable acknowledgment.	
1	Enable acknowledgment. During the ninth clock period, the SDAAn line is set to low level.	
Condition for clearing (ACKEn = 0)		Condition for setting (ACKEn = 1)
<ul style="list-style-type: none"> <li>• Cleared by instruction</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>

**Notes 1.** The signal of this bit is invalid while IICEn 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.

**Remark** n = 0

Figure 15-6. Format of IICA Control Register n0 (IICCTLn0) (3/4)

STTn <sup>Notes</sup> 1,2	Start condition trigger	
0	Do not generate a start condition.	
1	<p>When bus is released (in standby state, when IICBSYn = 0): If this bit is set (1), a start condition is generated (startup as the master).</p> <p>When a third party is communicating:</p> <ul style="list-style-type: none"> <li>• When communication reservation function is enabled (IICRSVn = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released.</li> <li>• When communication reservation function is disabled (IICRSVn = 1) Even if this bit is set (1), the STTn bit is cleared and the STTn clear flag (STCFn) is set (1). No start condition is generated.</li> </ul> <p>In the wait state (when master device): Generates a restart condition after releasing the wait.</p>	
<p>Cautions concerning set timing</p> <ul style="list-style-type: none"> <li>• For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception.</li> <li>• For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the wait period that follows output of the ninth clock.</li> <li>• Cannot be set to 1 at the same time as stop condition trigger (SPTn).</li> <li>• Once STTn is set (1), setting it again (1) before the clear condition is met is not allowed.</li> </ul>		
Condition for clearing (STTn = 0)		Condition for setting (STTn = 1)
<ul style="list-style-type: none"> <li>• Cleared by setting the STTn bit to 1 while communication reservation is prohibited.</li> <li>• Cleared by loss in arbitration</li> <li>• Cleared after start condition is generated by master device</li> <li>• Cleared by LRELn = 1 (exit from communications)</li> <li>• When IICEn = 0 (operation stop)</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>

- Notes**
1. The signal of this bit is invalid while IICEn is 0.
  2. The STTn bit is always read as 0.

- Remarks**
1. Bit 1 (STTn) becomes 0 when it is read after data setting.
  2. IICRSVn: Bit 0 of IIC flag register n (IICFn)  
STCFn: Bit 7 of IIC flag register n (IICFn)
  3. n = 0

**Figure 15-6. Format of IICA Control Register n0 (IICCTLn0) (4/4)**

SPTn <sup>Note</sup>	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer).
<p>Cautions concerning set timing</p> <ul style="list-style-type: none"> <li>• For master reception:     Cannot be set to 1 during transfer.                                                Can be set to 1 only in the waiting period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception.</li> <li>• For master transmission: A stop condition cannot be generated normally during the acknowledge period.                                                Therefore, set it during the wait period that follows output of the ninth clock.</li> <li>• Cannot be set to 1 at the same time as start condition trigger (STTn).</li> <li>• The SPTn bit can be set to 1 only when in master mode.</li> <li>• When the WTIMn bit has been cleared to 0, if the SPTn bit is set to 1 during the wait 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 WTIMn bit should be changed from 0 to 1 during the wait period following the output of eight clocks, and the SPTn bit should be set to 1 during the wait period that follows the output of the ninth clock.</li> <li>• Once SPTn is set (1), setting it again (1) before the clear condition is met is not allowed.</li> </ul>	
Condition for clearing (SPTn = 0)	Condition for setting (SPTn = 1)
<ul style="list-style-type: none"> <li>• Cleared by loss in arbitration</li> <li>• Automatically cleared after stop condition is detected</li> <li>• Cleared by LRELn = 1 (exit from communications)</li> <li>• When IICEn = 0 (operation stop)</li> <li>• Reset</li> </ul>	<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>

**Note** The STTn bit is always read as 0.

**Caution** When bit 3 (TRCn) of the IICA status register n (IICSn) is set to 1 (transmission status), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and wait is canceled, after which the TRCn bit is cleared (reception status) and the SDAAn line is set to high impedance. Release the wait performed while the TRCn bit is 1 (transmission status) by writing to the IICA shift register n.

**Remarks** 1. Bit 0 (SPTn) becomes 0 when it is read after data setting.  
 2. n = 0



**15.3.3 IICA status register n (IICSn)**

This register indicates the status of I<sup>2</sup>C.

The IICSn register is read by a 1-bit or 8-bit memory manipulation instruction only when STTn = 1 and during the wait period.

Reset signal generation clears this register to 00H.

**Caution** Reading the IICSn register while the address match wakeup function is enabled (WUPn = 1) in STOP mode is prohibited. When the WUPn bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICAn 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 (SPIEn = 1) the interrupt generated by detecting a stop condition and read the IICSn register after the interrupt has been detected.

**Remark** STTn: bit 1 of IICA control register n0 (IICCTLn0)  
 WUPn: bit 7 of IICA control register n1 (IICCTLn1)

**Figure 15-7. Format of IICA Status Register n (IICSn) (1/3)**

Address: FFF51H      After reset: 00H      R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTS <sub>n</sub>	ALD <sub>n</sub>	EXC <sub>n</sub>	COI <sub>n</sub>	TRC <sub>n</sub>	ACKD <sub>n</sub>	STD <sub>n</sub>	SPD <sub>n</sub>

MSTS <sub>n</sub>	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status
Condition for clearing (MSTS <sub>n</sub> = 0)	
<ul style="list-style-type: none"> <li>When a stop condition is detected</li> <li>When ALD<sub>n</sub> = 1 (arbitration loss)</li> <li>Cleared by LREL<sub>n</sub> = 1 (exit from communications)</li> <li>When the IICEn bit changes from 1 to 0 (operation stop)</li> <li>Reset</li> </ul>	
Condition for setting (MSTS <sub>n</sub> = 1)	
<ul style="list-style-type: none"> <li>When a start condition is generated</li> </ul>	

ALD <sub>n</sub>	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 MSTS <sub>n</sub> bit is cleared.
Condition for clearing (ALD <sub>n</sub> = 0)	
<ul style="list-style-type: none"> <li>Automatically cleared after the IICSn register is read<sup>Note</sup></li> <li>When the IICEn bit changes from 1 to 0 (operation stop)</li> <li>Reset</li> </ul>	
Condition for setting (ALD <sub>n</sub> = 1)	
<ul style="list-style-type: none"> <li>When the arbitration result is a "loss".</li> </ul>	

**Note** This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than the IICSn register. Therefore, when using the ALD<sub>n</sub> bit, read the data of this bit before the data of the other bits.

- Remarks**
- LREL<sub>n</sub>: Bit 6 of IICA control register n0 (IICCTLn0)  
 IICEn: Bit 7 of IICA control register n0 (IICCTLn0)
  - n = 0

Figure 15-7. Format of IICA Status Register n (IICSn) (2/3)

EXCn	Detection of extension code reception	
0	Extension code was not received.	
1	Extension code was received.	
Condition for clearing (EXCn = 0)		Condition for setting (EXCn = 1)
<ul style="list-style-type: none"> <li>When a start condition is detected</li> <li>When a stop condition is detected</li> <li>Cleared by LRELn = 1 (exit from communications)</li> <li>When the IICEn bit changes from 1 to 0 (operation stop)</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>When the higher four bits of the received address data is either "0000" or "1111" (set at the rising edge of the eighth clock).</li> </ul>

COIn	Detection of matching addresses	
0	Addresses do not match.	
1	Addresses match.	
Condition for clearing (COIn = 0)		Condition for setting (COIn = 1)
<ul style="list-style-type: none"> <li>When a start condition is detected</li> <li>When a stop condition is detected</li> <li>Cleared by LRELn = 1 (exit from communications)</li> <li>When the IICEn bit changes from 1 to 0 (operation stop)</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>When the received address matches the local address (slave address register n (SVAn)) (set at the rising edge of the eighth clock).</li> </ul>

TRCn	Detection of transmit/receive status	
0	Receive status (other than transmit status). The SDAAn line is set for high impedance.	
1	Transmit status. The value in the SOn latch is enabled for output to the SDAAn line (valid starting at the falling edge of the first byte's ninth clock).	
Condition for clearing (TRCn = 0)		Condition for setting (TRCn = 1)
<p>&lt;Both master and slave&gt;</p> <ul style="list-style-type: none"> <li>When a stop condition is detected</li> <li>Cleared by LRELn = 1 (exit from communications)</li> <li>When the IICEn bit changes from 1 to 0 (operation stop)</li> <li>Cleared by WRELn = 1<sup>Note</sup> (wait cancel)</li> <li>When the ALDn bit changes from 0 to 1 (arbitration loss)</li> <li>Reset</li> <li>When not used for communication (MSTS<sub>n</sub>, EXC<sub>n</sub>, COIn = 0)</li> </ul> <p>&lt;Master&gt;</p> <ul style="list-style-type: none"> <li>When "1" is output to the first byte's LSB (transfer direction specification bit)</li> </ul> <p>&lt;Slave&gt;</p> <ul style="list-style-type: none"> <li>When a start condition is detected</li> <li>When "0" is input to the first byte's LSB (transfer direction specification bit)</li> </ul>		<p>&lt;Master&gt;</p> <ul style="list-style-type: none"> <li>When a start condition is generated</li> <li>When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer)</li> </ul> <p>&lt;Slave&gt;</p> <ul style="list-style-type: none"> <li>When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)</li> </ul>

**Note** When bit 3 (TRCn) of the IICA status register n (IICSn) is set to 1 (transmission status), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and wait is canceled, after which the TRCn bit is cleared (reception status) and the SDAAn line is set to high impedance. Release the wait performed while the TRCn bit is 1 (transmission status) by writing to the IICA shift register n.

- Remarks**
- LRELn: Bit 6 of IICA control register n0 (IICCTLn0)  
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)
  - n = 0

Figure 15-7. Format of IICA Status Register n (IICS<sub>n</sub>) (3/3)

ACKD <sub>n</sub>	Detection of acknowledge (ACK)	
0	Acknowledge was not detected.	
1	Acknowledge was detected.	
Condition for clearing (ACKD <sub>n</sub> = 0)		Condition for setting (ACKD <sub>n</sub> = 1)
<ul style="list-style-type: none"> <li>When a stop condition is detected</li> <li>At the rising edge of the next byte's first clock</li> <li>Cleared by LREL<sub>n</sub> = 1 (exit from communications)</li> <li>When the IICEn bit changes from 1 to 0 (operation stop)</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>After the SDAAn line is set to low level at the rising edge of SCLAn line's ninth clock</li> </ul>

STD <sub>n</sub>	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 (STD <sub>n</sub> = 0)		Condition for setting (STD <sub>n</sub> = 1)
<ul style="list-style-type: none"> <li>When a stop condition is detected</li> <li>At the rising edge of the next byte's first clock following address transfer</li> <li>Cleared by LREL<sub>n</sub> = 1 (exit from communications)</li> <li>When the IICEn bit changes from 1 to 0 (operation stop)</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>When a start condition is detected</li> </ul>

SPD <sub>n</sub>	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 (SPD <sub>n</sub> = 0)		Condition for setting (SPD <sub>n</sub> = 1)
<ul style="list-style-type: none"> <li>At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition</li> <li>When the WUP<sub>n</sub> bit changes from 1 to 0</li> <li>When the IICEn bit changes from 1 to 0 (operation stop)</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>When a stop condition is detected</li> </ul>

- Remarks 1.** LREL<sub>n</sub>: Bit 6 of IICA control register n0 (IICCTLn0)  
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)
- 2.** n = 0

### 15.3.4 IICA flag register n (IICFn)

This register sets the operation mode of I<sup>2</sup>C and indicates the status of the I<sup>2</sup>C bus.

The IICFn register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STTn clear flag (STCFn) and I<sup>2</sup>C bus status flag (IICBSYn) bits are read-only.

The IICRSVn bit can be used to enable/disable the communication reservation function.

The STCENn bit can be used to set the initial value of the IICBSYn bit.

The IICRSVn and STCENn bits can be written only when the operation of I<sup>2</sup>C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) = 0). When operation is enabled, the IICFn register can be read.

Reset signal generation clears this register to 00H.

**Figure 15-8. Format of IICA Flag Register n (IICFn)**

Address: FFF52H    After reset: 00H    R/W<sup>Note</sup>

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn clear flag	
0	Generate start condition	
1	Start condition generation unsuccessful: clear the STTn flag	
Condition for clearing (STCFn = 0)		Condition for setting (STCFn = 1)
<ul style="list-style-type: none"> <li>Cleared by STTn = 1</li> <li>When IICEn = 0 (operation stop)</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>Generating start condition unsuccessful and the STTn bit cleared to 0 when communication reservation is disabled (IICRSVn = 1).</li> </ul>

IICBSYn	I <sup>2</sup> C bus status flag	
0	Bus release status (communication initial status when STCENn = 1)	
1	Bus communication status (communication initial status when STCENn = 0)	
Condition for clearing (IICBSYn = 0)		Condition for setting (IICBSYn = 1)
<ul style="list-style-type: none"> <li>Detection of stop condition</li> <li>When IICEn = 0 (operation stop)</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>Detection of start condition</li> <li>Setting of the IICEn bit when STCENn = 0</li> </ul>

STCENn	Initial start enable trigger	
0	After operation is enabled (IICEn = 1), enable generation of a start condition upon detection of a stop condition.	
1	After operation is enabled (IICEn = 1), enable generation of a start condition without detecting a stop condition.	
Condition for clearing (STCENn = 0)		Condition for setting (STCENn = 1)
<ul style="list-style-type: none"> <li>Cleared by instruction</li> <li>Detection of start condition</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>Set by instruction</li> </ul>

IICRSVn	Communication reservation function disable bit	
0	Enable communication reservation	
1	Disable communication reservation	
Condition for clearing (IICRSVn = 0)		Condition for setting (IICRSVn = 1)
<ul style="list-style-type: none"> <li>Cleared by instruction</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>Set by instruction</li> </ul>

**Note** Bits 6 and 7 are read-only.

- Cautions**
1. Write to the STCENn bit only when the operation is stopped (IICEn = 0).
  2. As the bus release status (IICBSYn = 0) is recognized regardless of the actual bus status when STCENn = 1, when generating the first start condition (STTn = 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 IICRSVn only when the operation is stopped (IICEn = 0).

- Remarks**
1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)  
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)
  2. n = 0

**15.3.5 IICA control register n1 (IICCTLn1)**

This register is used to set the operation mode of I<sup>2</sup>C and detect the statuses of the SCLAn and SDAAn pins.

The IICCTLn1 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLDn and DADn bits are read-only.

Set the IICCTLn1 register, except the WUPn bit, while operation of I<sup>2</sup>C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation clears this register to 00H.

**Figure 15-9. Format of IICA Control Register n1 (IICCTLn1) (1/2)**

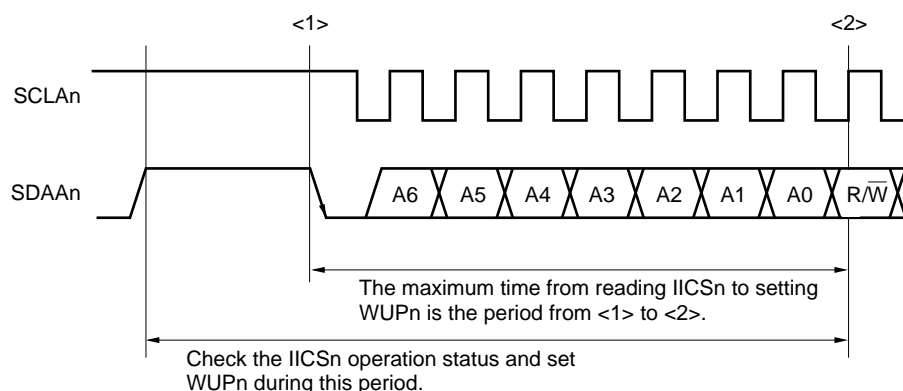
Address: F0231H    After reset: 00H    R/W<sup>Note 1</sup>

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
IICCTLn1	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

WUPn	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.
<p>To shift to STOP mode when WUPn = 1, execute the STOP instruction at least three clocks of f<sub>MCK</sub> after setting (1) the WUPn bit (see <b>Figure 15-22 Flow When Setting WUPn = 1</b>).</p> <p>Clear (0) the WUPn bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUPn bit. (The wait must be released and transmit data must be written after the WUPn bit has been cleared (0).)</p> <p>The interrupt timing when the address has matched or when an extension code has been received, while WUPn = 1, is identical to the interrupt timing when WUPn = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUPn = 1, a stop condition interrupt is not generated even if the SPIEn bit is set to 1.</p>	
Condition for clearing (WUPn = 0)	Condition for setting (WUPn = 1)
<ul style="list-style-type: none"> <li>• Cleared by instruction (after address match or extension code reception)</li> </ul>	<ul style="list-style-type: none"> <li>• Set by instruction (when the MSTSn, EXCn, and COIn bits are "0", and the STDn bit also "0" (communication not entered))<sup>Note 2</sup></li> </ul>

**Notes 1.** Bits 4 and 5 are read-only.

**2.** The status of the IICA status register n (IICSn) must be checked and the WUPn bit must be set during the period shown below.



**Remark** n = 0

Figure 15-9. Format of IICA Control Register n1 (IICCTLn1) (2/2)

CLDn	Detection of SCLAn pin level (valid only when IICEn = 1)	
0	The SCLAn pin was detected at low level.	
1	The SCLAn pin was detected at high level.	
Condition for clearing (CLDn = 0)		Condition for setting (CLDn = 1)
<ul style="list-style-type: none"> <li>When the SCLAn pin is at low level</li> <li>When IICEn = 0 (operation stop)</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>When the SCLAn pin is at high level</li> </ul>

DADn	Detection of SDAAn pin level (valid only when IICEn = 1)	
0	The SDAAn pin was detected at low level.	
1	The SDAAn pin was detected at high level.	
Condition for clearing (DADn = 0)		Condition for setting (DADn = 1)
<ul style="list-style-type: none"> <li>When the SDAAn pin is at low level</li> <li>When IICEn = 0 (operation stop)</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>When the SDAAn pin is at high level</li> </ul>

SMCn	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).	

DFCn	Digital filter operation control	
0	Digital filter off.	
1	Digital filter on.	
<p>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).          The digital filter is used for noise elimination in fast mode and fast mode plus.</p>		

PRSn	IICA operation clock ( $f_{MCK}$ ) control	
0	Selects $f_{CLK}$ ( $1 \text{ MHz} \leq f_{CLK} \leq 20 \text{ MHz}$ )	
1	Selects $f_{CLK}/2$ ( $20 \text{ MHz} < f_{CLK}$ )	

- Cautions**
- The fastest operation frequency of the IICA operation clock ( $f_{MCK}$ ) is 20 MHz (Max.). Set bit 0 (PRSn) of the IICA control register n1 (IICCTLn1) to "1" only when the  $f_{CLK}$  exceeds 20 MHz.
  - 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:  $f_{CLK} = 3.5 \text{ MHz (MIN.)}$
    - Fast mode plus:  $f_{CLK} = 10 \text{ MHz (MIN.)}$
    - Normal mode:  $f_{CLK} = 1 \text{ MHz (MIN.)}$
  - The fast mode plus is only available in the products for A: Consumer applications ( $T_A = -40 \text{ }^\circ\text{C}$  to  $+85 \text{ }^\circ\text{C}$ ).

- Remarks**
- IICEn: Bit 7 of IICA control register n0 (IICCTLn0)
  - n = 0

### 15.3.6 IICA low-level width setting register n (IICWLn)

This register is used to set the low-level width ( $t_{LOW}$ ) of the SCLAn 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 IICWLn register can be set by an 8-bit memory manipulation instruction.

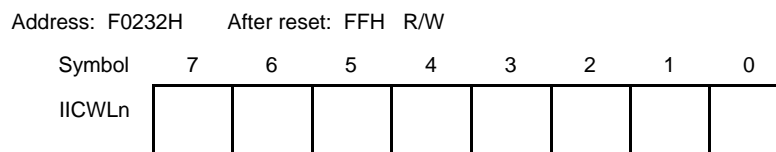
Set the IICWLn register while operation of I<sup>2</sup>C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

For details about setting the IICWLn register, see **15.4.2 Setting transfer clock by using IICWLn and IICWHn registers.**

<R> The data hold time is one-quarter of the time set by the IICWLn register.

**Figure 15-10. Format of IICA Low-Level Width Setting Register n (IICWLn)**



### 15.3.7 IICA high-level width setting register n (IICWHn)

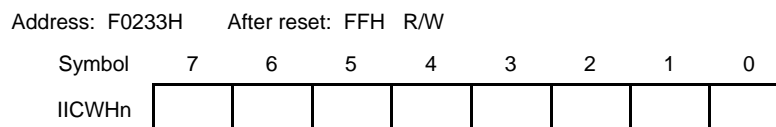
This register is used to set the high-level width of the SCLAn pin signal that is output by serial interface IICA.

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

Set the IICWHn register while operation of I<sup>2</sup>C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

**Figure 15-11. Format of IICA High-Level Width Setting Register n (IICWHn)**



- Remarks 1.** For setting procedures of the transfer clock on master side and of the IICWLn and IICWHn registers on slave side, see **15.4.2 (1)** and **15.4.2 (2)**, respectively.
- 2.** n = 0

### 15.3.8 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 IICEn bit (bit 7 of IICA control register n0 (IICCTLn0)) to 1 before setting the output mode because the P60/SCLA0 and P61/SDAA0 pins output a low level (fixed) when the IICEn 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 15-12. 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)



### 15.4 I<sup>2</sup>C Bus Mode Functions

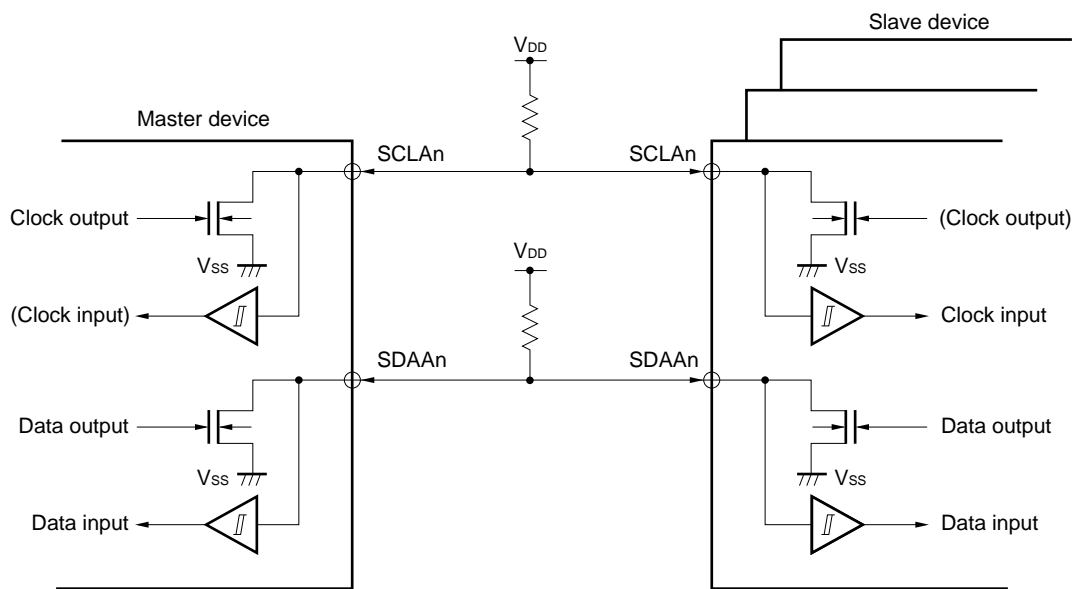
#### 15.4.1 Pin configuration

The serial clock pin (SCLAn) and the serial data bus pin (SDAAn) are configured as follows.

- (1) SCLAn .... 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) SDAAn .... 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 15-13. Pin Configuration Diagram**



**Remark** n = 0

### 15.4.2 Setting transfer clock by using IICWLn and IICWHn registers

#### (1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{\text{MCK}}}{\text{IICWL0} + \text{IICWH0} + f_{\text{MCK}}(t_{\text{R}} + t_{\text{F}})}$$

At this time, the optimal setting values of the IICWLn and IICWHn registers are as follows.  
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$\begin{aligned} \text{IICWLn} &= \frac{0.52}{\text{Transfer clock}} \times f_{\text{MCK}} \\ \text{IICWHn} &= \left( \frac{0.48}{\text{Transfer clock}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}} \end{aligned}$$

- When the normal mode

$$\begin{aligned} \text{IICWLn} &= \frac{0.47}{\text{Transfer clock}} \times f_{\text{MCK}} \\ \text{IICWHn} &= \left( \frac{0.53}{\text{Transfer clock}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}} \end{aligned}$$

- When the fast mode plus

$$\begin{aligned} \text{IICWLn} &= \frac{0.50}{\text{Transfer clock}} \times f_{\text{MCK}} \\ \text{IICWHn} &= \left( \frac{0.50}{\text{Transfer clock}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}} \end{aligned}$$

#### (2) Setting IICWLn and IICWHn registers on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$\begin{aligned} \text{IICWLn} &= 1.3 \mu\text{S} \times f_{\text{MCK}} \\ \text{IICWHn} &= (1.2 \mu\text{S} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}} \end{aligned}$$

- When the normal mode

$$\begin{aligned} \text{IICWLn} &= 4.7 \mu\text{S} \times f_{\text{MCK}} \\ \text{IICWHn} &= (5.3 \mu\text{S} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}} \end{aligned}$$

- When the fast mode plus

$$\begin{aligned} \text{IICWLn} &= 0.50 \mu\text{S} \times f_{\text{MCK}} \\ \text{IICWHn} &= (0.50 \mu\text{S} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}} \end{aligned}$$

(Cautions and Remarks are listed on the next page.)

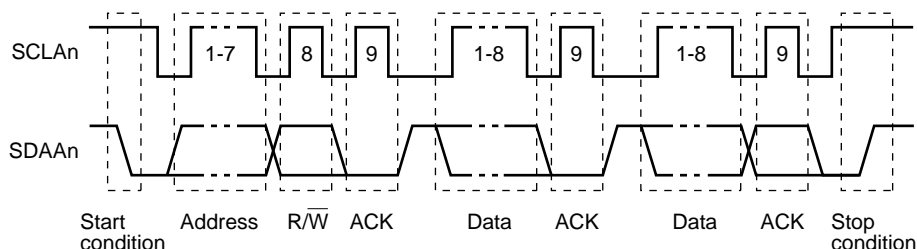
- Cautions**
- 1. The fastest operation frequency of the IICA operation clock ( $f_{MCK}$ ) is 20 MHz (Max.).  
Set bit 0 (PRSn) of the IICA control register n1 (IICCTLn1) to "1" only when the  $f_{CLK}$  exceeds 20 MHz.**
  - 2. Note the minimum  $f_{CLK}$  operation frequency when setting the transfer 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 SDAAn and SCLAn signals separately, because they differ depending on the pull-up resistance and wire load.**
  - 2. IICWLn: IICA low-level width setting register n**  
**IICWHn: IICA high-level width setting register n**  
 **$t_F$ : SDAAn and SCLAn signal falling times**  
 **$t_R$ : SDAAn and SCLAn signal rising times**  
 **$f_{MCK}$ : IICA operation clock frequency**
  - 3.  $n = 0$**

### 15.5 I<sup>2</sup>C Bus Definitions and Control Methods

The following section describes the I<sup>2</sup>C bus's serial data communication format and the signals used by the I<sup>2</sup>C bus. Figure 15-14 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I<sup>2</sup>C bus's serial data bus.

**Figure 15-14. I<sup>2</sup>C Bus Serial Data Transfer Timing**



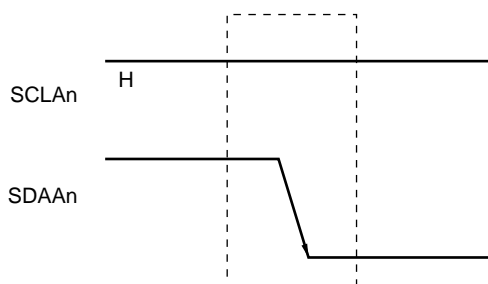
The master device generates the start condition, slave address, and stop condition. The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLAn) is continuously output by the master device. However, in the slave device, the SCLAn pin low level period can be extended and a wait can be inserted.

#### 15.5.1 Start conditions

A start condition is met when the SCLAn pin is at high level and the SDAAn pin changes from high level to low level. The start conditions for the SCLAn pin and SDAAn 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 15-15. Start Conditions**



A start condition is output when bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set (1) after a stop condition has been detected (SPDn: Bit 0 of the IICA status register n (IICSn) = 1). When a start condition is detected, bit 1 (STDn) of the IICSn register is set (1).

**Remark** n = 0

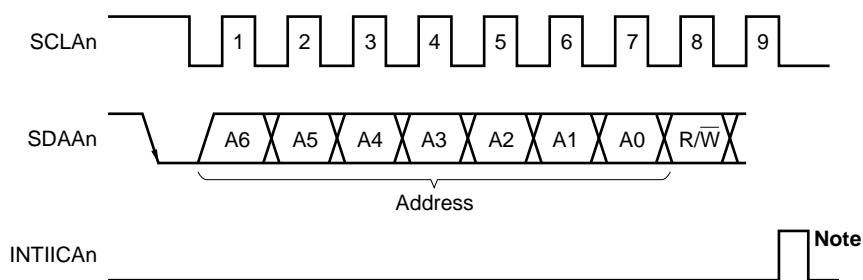
### 15.5.2 Addresses

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 n (SVAn). If the address data matches the SVAn 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 15-16. Address



**Note** INTIICAn 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 **15.5.3 Transfer direction specification** are written to the IICAn shift register n (IICAn). The received addresses are written to the IICAn register.

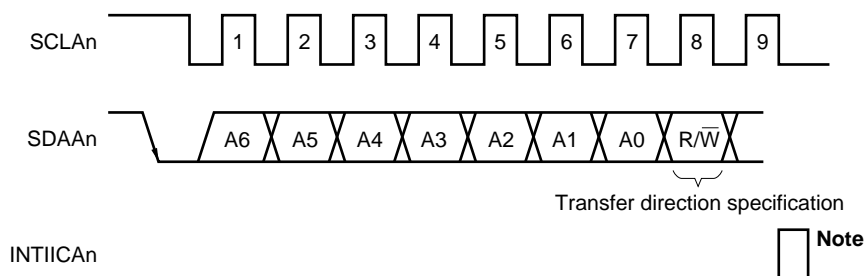
The slave address is assigned to the higher 7 bits of the IICAn register.

### 15.5.3 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 15-17. Transfer Direction Specification



**Note** INTIICAn is not issued if data other than a local address or extension code is received during slave device operation.

**Remark** n = 0

### 15.5.4 Acknowledge (ACK)

ACK is used to check the status of serial data at the transmission and reception sides.

The reception side returns ACK each time it has received 8-bit data.

The transmission side usually receives ACK after transmitting 8-bit data. When ACK is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether ACK has been detected can be checked by using bit 2 (ACKDn) of the IICA status register n (IICSn).

When the master receives the last data item, it does not return ACK and instead generates a stop condition. If a slave does not return ACK after receiving data, the master outputs a stop condition or restart condition and stops transmission. If ACK is not returned, the possible causes are as follows.

- <1> Reception was not performed normally.
- <2> The final data item was received.
- <3> The reception side specified by the address does not exist.

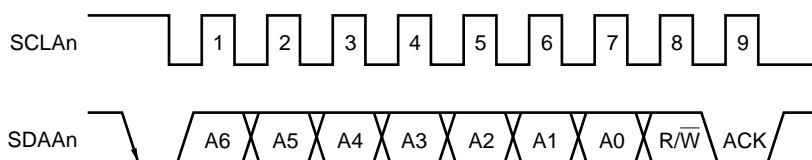
To generate ACK, the reception side makes the SDAAn line low at the ninth clock (indicating normal reception).

Automatic generation of ACK is enabled by setting bit 2 (ACKEn) of IICA control register n0 (IICCTLn0) to 1. Bit 3 (TRCn) of the IICSn register is set by the data of the eighth bit that follows 7-bit address information. Usually, set the ACKEn bit to 1 for reception (TRCn = 0).

If a slave can receive no more data during reception (TRCn = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKEn bit to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRCn = 0), it must clear the ACKEn bit to 0 so that ACK is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 15-18. ACK



When the local address is received, ACK is automatically generated, regardless of the value of the ACKEn bit. When an address other than that of the local address is received, ACK is not generated (NACK).

When an extension code is received, ACK is generated if the ACKEn bit is set to 1 in advance.

How ACK is generated when data is received differs as follows depending on the setting of the wait timing.

- When 8-clock wait state is selected (bit 3 (WTIMn) of IICCTLn0 register = 0):  
By setting the ACKEn bit to 1 before releasing the wait state, ACK is generated at the falling edge of the eighth clock of the SCLAn pin.
- When 9-clock wait state is selected (bit 3 (WTIMn) of IICCTLn0 register = 1):  
ACK is generated by setting the ACKEn bit to 1 in advance.

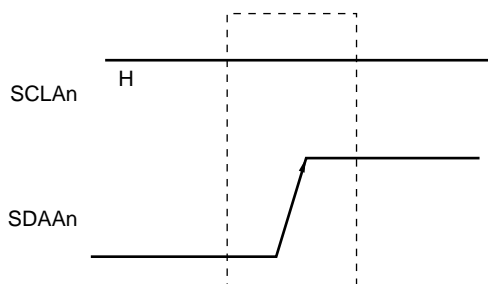
**Remark** n = 0

### 15.5.5 Stop condition

When the SCLAn pin is at high level, changing the SDAAn 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 15-19. Stop Condition**



A stop condition is generated when bit 0 (SPTn) of IICA control register n0 (IICCTLn0) is set to 1. When the stop condition is detected, bit 0 (SPDn) of the IICA status register n (IICSn) is set to 1 and INTIICAn is generated when bit 4 (SPIEn) of the IICCTLn0 register is set to 1.

**Remark** n = 0

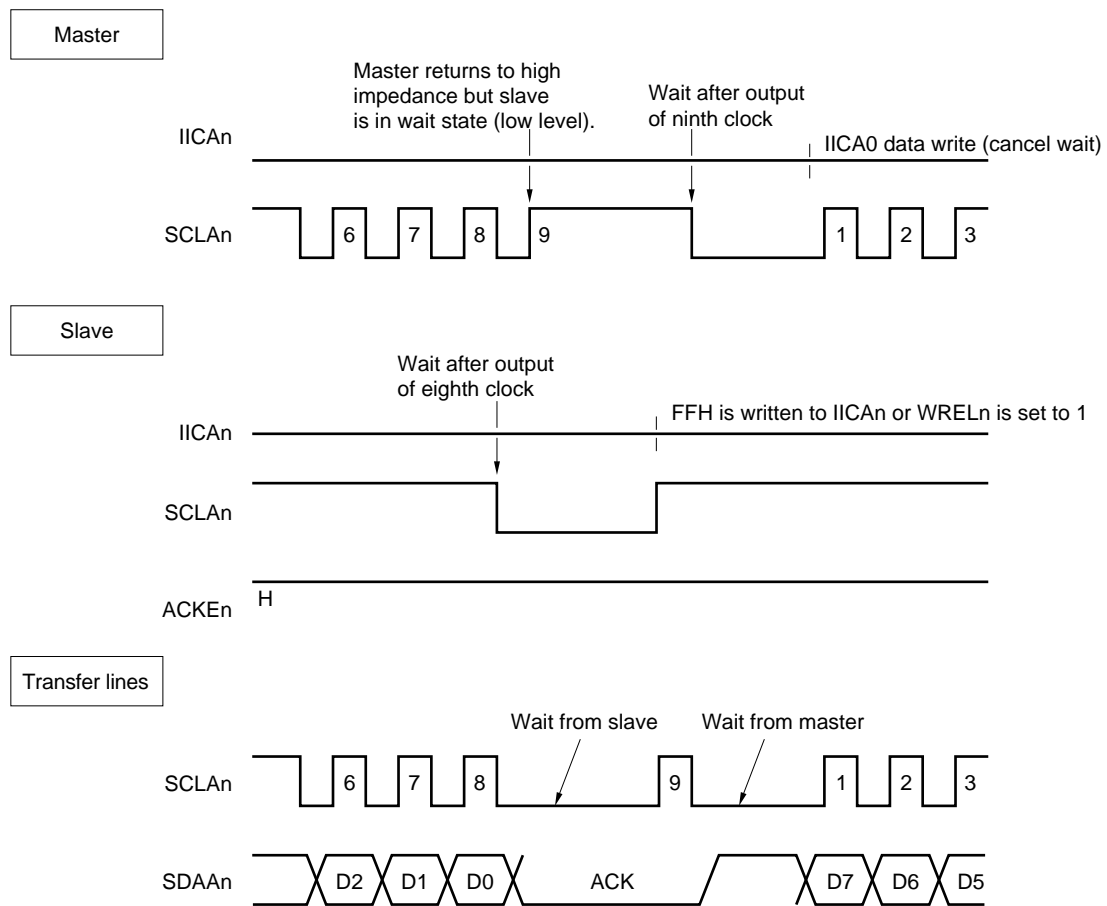
**15.5.6 Wait**

The wait 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 wait state).

Setting the SCLAn pin to low level notifies the communication partner of the wait state. When wait state has been canceled for both the master and slave devices, the next data transfer can begin.

**Figure 15-20. Wait (1/2)**

**(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKEn = 1)**

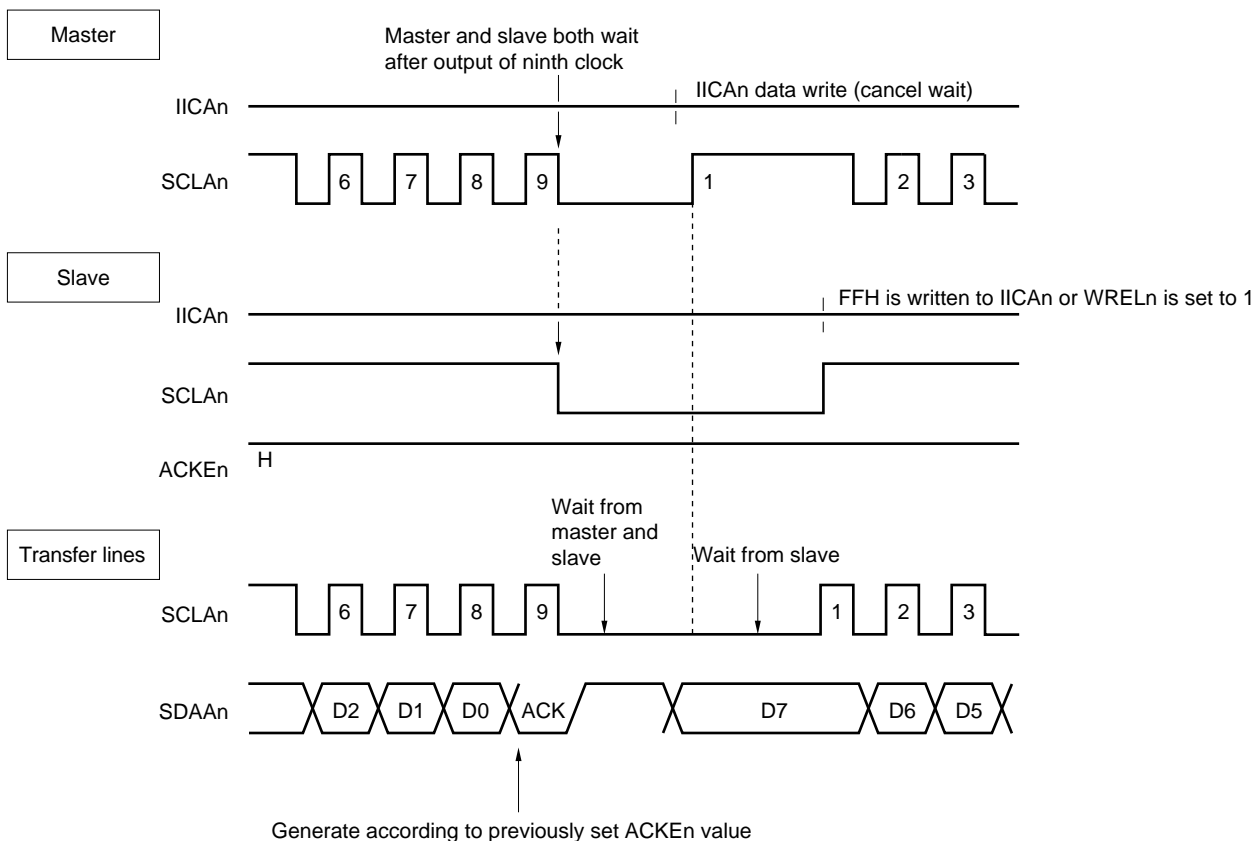


**Remark** n = 0



Figure 15-20. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait  
(master transmits, slave receives, and ACKEn = 1)



**Remark** ACKEn: Bit 2 of IICA control register n0 (IICCTLn0)  
WRELn: Bit 5 of IICA control register n0 (IICCTLn0)

A wait may be automatically generated depending on the setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0). Normally, the receiving side cancels the wait state when bit 5 (WRELn) of the IICCTLn0 register is set to 1 or when FFH is written to the IICA shift register n (IICAn), and the transmitting side cancels the wait state when data is written to the IICAn register.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STTn) of the IICCTLn0 register to 1
- By setting bit 0 (SPTn) of the IICCTLn0 register to 1

**Remark** n = 0

### 15.5.7 Canceling wait

The I<sup>2</sup>C usually cancels a wait state by the following processing.

- Writing data to the IICA shift register n (IICAn)
- Setting bit 5 (WRELn) of IICA control register n0 (IICCTLn0) (canceling wait)
- Setting bit 1 (STTn) of the IICCTLn0 register (generating start condition)<sup>Note</sup>
- Setting bit 0 (SPTn) of the IICCTLn0 register (generating stop condition)<sup>Note</sup>

**Note** Master only

When the above wait canceling processing is executed, the I<sup>2</sup>C cancels the wait state and communication is resumed.

To cancel a wait state and transmit data (including addresses), write the data to the IICAn register.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WRELn) of the IICCTLn0 register to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STTn) of the IICCTLn0 register to 1.

To generate a stop condition after canceling a wait state, set bit n (SPTn) of the IICCTLn0 register to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to the IICAn register after canceling a wait state by setting the WRELn bit to 1, an incorrect value may be output to SDAAn line because the timing for changing the SDAAn line conflicts with the timing for writing the IICAn register.

In addition to the above, communication is stopped if the IICEn bit is cleared to 0 when communication has been aborted, so that the wait state can be canceled.

If the I<sup>2</sup>C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LRELn) of the IICCTLn0 register, so that the wait state can be canceled.

**Caution** If a processing to cancel a wait state is executed when WUPn = 1, the wait state will not be canceled.

**Remark** n = 0

### 15.5.8 Interrupt request (INTIICAn) generation timing and wait control

The setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0) determines the timing by which INTIICAn is generated and the corresponding wait control, as shown in Table 15-2.

**Table 15-2. INTIICAn Generation Timing and Wait Control**

WTIMn	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	g <sup>Notes 1, 2</sup>	g <sup>Note 2</sup>	g <sup>Note 2</sup>	9	8	8
1	g <sup>Notes 1, 2</sup>	g <sup>Note 2</sup>	g <sup>Note 2</sup>	9	9	9

**Notes 1.** The slave device's INTIICAn signal and wait 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 n (SVAn).

At this point, ACK is generated regardless of the value set to the IICCTLn0 register's bit 2 (ACKEn). For a slave device that has received an extension code, INTIICAn occurs at the falling edge of the eighth clock.

However, if the address does not match after restart, INTIICAn is generated at the falling edge of the 9th clock, but wait does not occur.

**2.** If the received address does not match the contents of the slave address register n (SVAn) and extension code is not received, neither INTIICAn nor a wait occurs.

**Remark** The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

#### (1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIMn bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIMn bit.

#### (2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIMn bit.

#### (3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIMn bit.

#### (4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to the IICA shift register n (IICAn)
- Setting bit 5 (WRELn) of IICA control register n0 (IICCTLn0) (canceling wait)
- Setting bit 1 (STTn) of IICCTLn0 register (generating start condition)<sup>Note</sup>
- Setting bit 0 (SPTn) of IICCTLn0 register (generating stop condition)<sup>Note</sup>

**Note** Master only.

When an 8-clock wait has been selected (WTIMn = 0), the presence/absence of ACK generation must be determined prior to wait cancellation.

#### (5) Stop condition detection

INTIICAn is generated when a stop condition is detected (only when SPIEn = 1).

**Remark** n = 0

**15.5.9 Address match detection method**

In I<sup>2</sup>C 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 (INTIICAn) occurs when the address set to the slave address register n (SVAn) matches the slave address sent by the master device, or when an extension code has been received.

**15.5.10 Error detection**

In I<sup>2</sup>C bus mode, the status of the serial data bus (SDAAn) during data transmission is captured by the IICA shift register n (IICAn) 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.

**Remark** n = 0

**15.5.11 Extension code**

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXCn) is set to 1 for extension code reception and an interrupt request (INTIICAn) is issued at the falling edge of the eighth clock. The local address stored in the slave address register n (SVAn) 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 SVAn register is set to 11110xx0. Note that INTIICAn occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXCn = 1
- Seven bits of data match: COIn = 1

**Remark** EXCn: Bit 5 of IICA status register n (IICSn)  
 COIn: Bit 4 of IICA status register n (IICSn)

(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 (LRELn) of IICA control register n0 (IICCTLn0) to 1 to set the standby mode for the next communication operation.

**Table 15-3. 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)

**Remarks 1.** See the I<sup>2</sup>C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

**2.** n = 0

**15.5.12 Arbitration**

When several master devices simultaneously generate a start condition (when the STTn bit is set to 1 before the STDn 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.

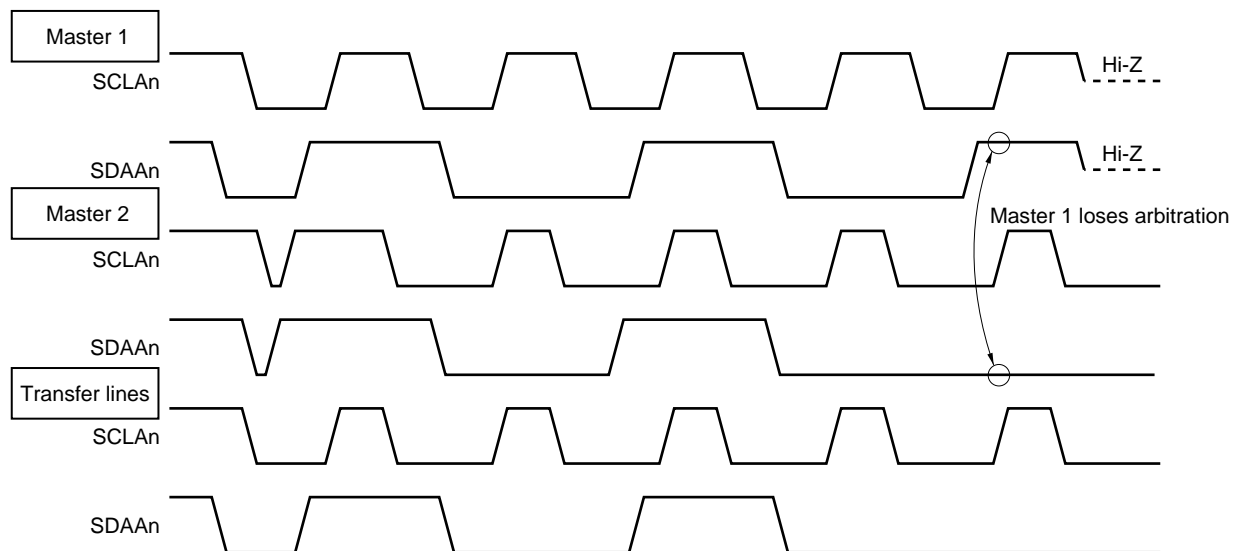
When one of the master devices loses in arbitration, an arbitration loss flag (ALDn) in the IICA status register n (IICSn) is set (1) via the timing by which the arbitration loss occurred, and the SCLAn and SDAAn lines are both set to high impedance, which releases the bus.

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 ALDn = 1 setting that has been made by software.

For details of interrupt request timing, see **15.5.8 Interrupt request (INTIICAn) generation timing and wait control.**

**Remark** STDn: Bit 1 of IICA status register n (IICSn)  
 STTn: Bit 1 of IICA control register n0 (IICCTLn0)

**Figure 15-21. Arbitration Timing Example**



**Remark** n = 0

**Table 15-4. 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 <sup>Note 1</sup>
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During ACK transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when SPIEn = 1) <sup>Note 2</sup>
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer <sup>Note 1</sup>
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIEn = 1) <sup>Note 2</sup>
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer <sup>Note 1</sup>
When SCLAn is at low level while attempting to generate a restart condition	

**Notes 1.** When the WTIMn bit (bit 3 of IICA control register n0 (IICCTLn0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIMn = 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 SPIEn = 1 for master device operation.

**Remarks 1.** SPIEn: Bit 4 of IICA control register n0 (IICCTLn0)

**2.** n = 0

**15.5.13 Wakeup function**

The I<sup>2</sup>C bus slave function is a function that generates an interrupt request signal (INTIICAn) when a local address and extension code have been received.

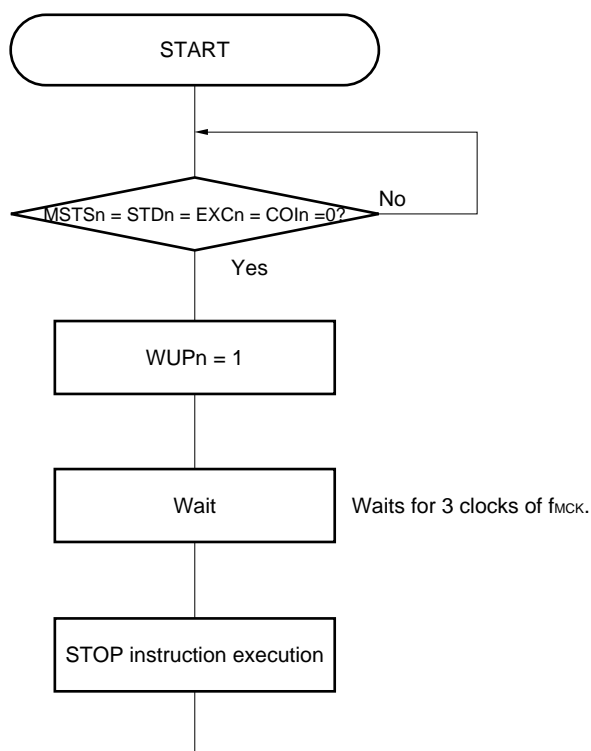
This function makes processing more efficient by preventing unnecessary INTIICAn signal from occurring when addresses do not match.

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 WUPn bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICAn) 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 WUPn bit after this interrupt has been generated.

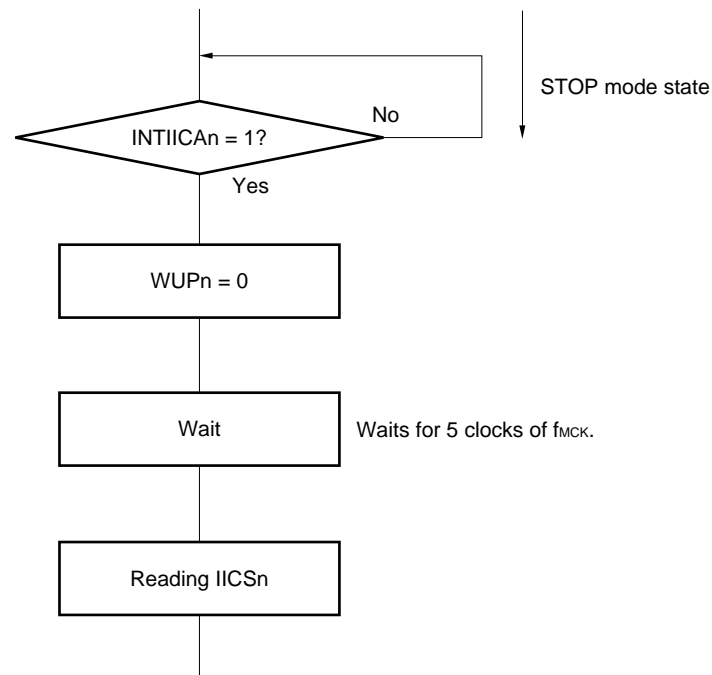
Figure 15-22 shows the flow for setting WUPn = 1 and Figure 15-23 shows the flow for setting WUPn = 0 upon an address match.

**Figure 15-22. Flow When Setting WUPn = 1**



**Remark** n = 0

Figure 15-23. Flow When Setting WUPn = 0 upon Address Match (Including Extension Code Reception)



Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

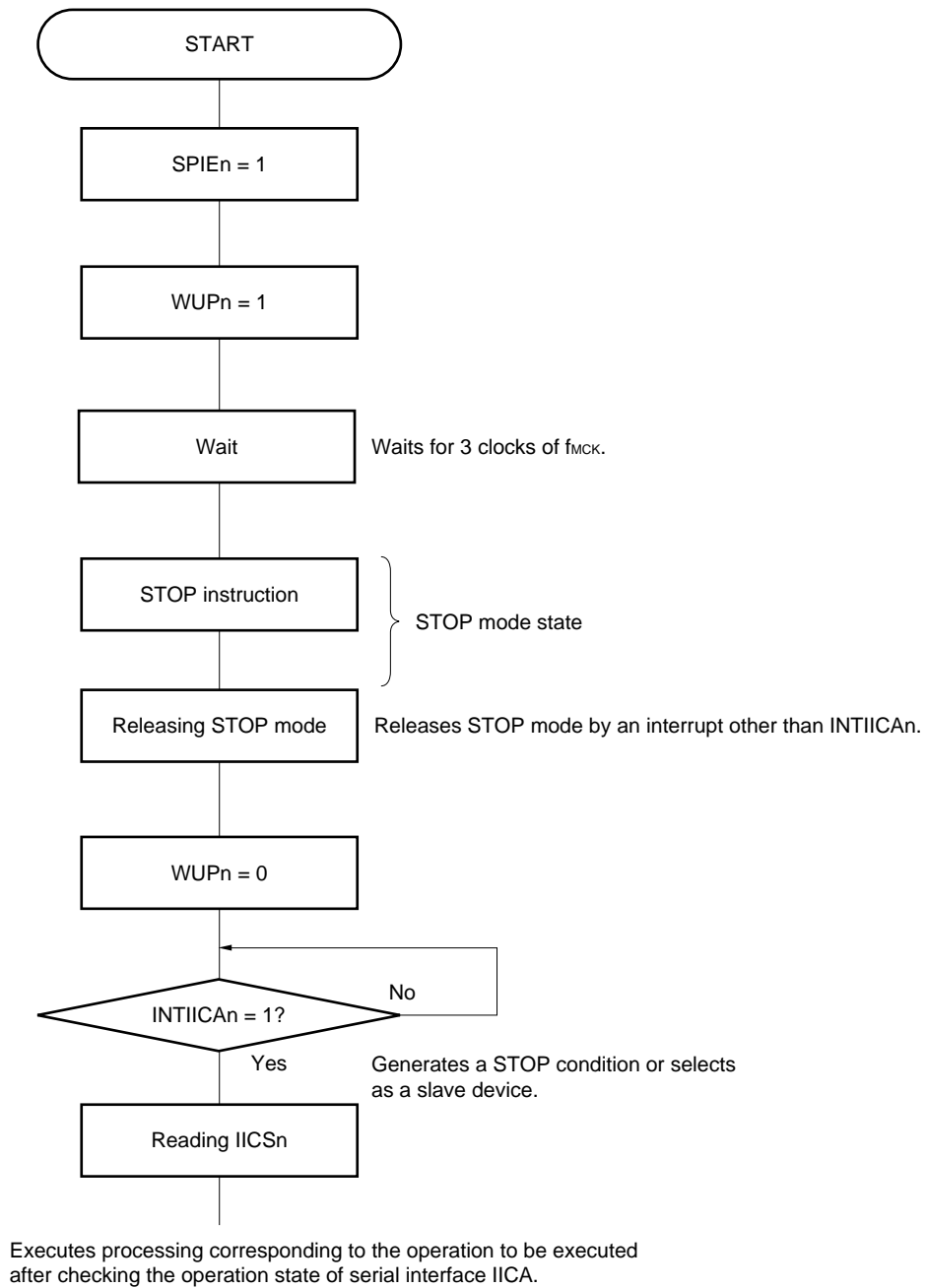
Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICAn) generated from serial interface IICA.

- When operating next IIC communication as master: Flow shown in Figure 15-24
- When operating next IIC communication as slave:
  - When restored by INTIICAn interrupt: Same as the flow in Figure 15-23
  - When restored by other than INTIICAn interrupt: Until the INTIICAn interrupt occurs, continue operating with WUPn left set to 1

**Remark** n = 0



Figure 15-24. When Operating as Master Device after Releasing STOP Mode Other than by INTIICAn



**Remark** n = 0

**15.5.14 Communication reservation**

**(1) When communication reservation function is enabled (bit n (IICRSVn) of IICA flag register n (IICFn) = 0)**

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LRELn) of IICA control register n0 (IICCTLn0) to 1 and saving communication).

If bit 1 (STTn) of the IICCTLn0 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 n (IICAn) after bit 4 (SPIEn) of the IICCTLn0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICAn) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICAn register before the stop condition is detected is invalid.

When the STTn 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 MSTSn bit (bit 7 of the IICA status register n (IICSn)) after the STTn bit is set to 1 and the wait time elapses.

Use software to secure the wait time calculated by the following expression.

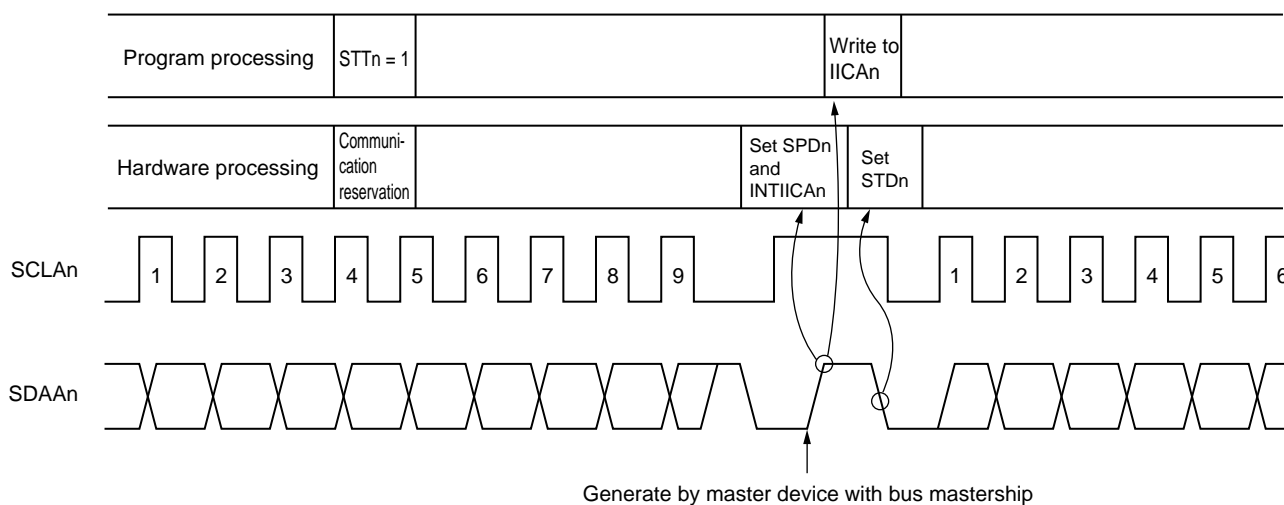
<R>

Wait time from setting STTn = 1 to checking the MSTSn flag:  
 $(IICWLn \text{ setting value} + IICWHn \text{ setting value} + 4) / f_{MCK} + t_F \times 2$

- Remarks**
1. IICWLn: IICA low-level width setting register n  
 IICWHn: IICA high-level width setting register n  
 t<sub>F</sub>: SDAAn and SCLAn signal falling times  
 f<sub>MCK</sub>: IICA operation clock frequency
  2. n = 0

Figure 15-25 shows the communication reservation timing.

**Figure 15-25. Communication Reservation Timing**



- Remark** IICAn: IICA shift register n
- STTn: Bit 1 of IICA control register n0 (IICCTLn0)
- STDn: Bit 1 of IICA status register n (IICSn)
- SPDn: Bit 0 of IICA status register n (IICSn)

Communication reservations are accepted via the timing shown in Figure 15-26. After bit 1 (STDn) of the IICA status register n (IICSn) is set to 1, a communication reservation can be made by setting bit 1 (STTn) of IICA control register n0 (IICCTLn0) to 1 before a stop condition is detected.

**Figure 15-26. Timing for Accepting Communication Reservations**

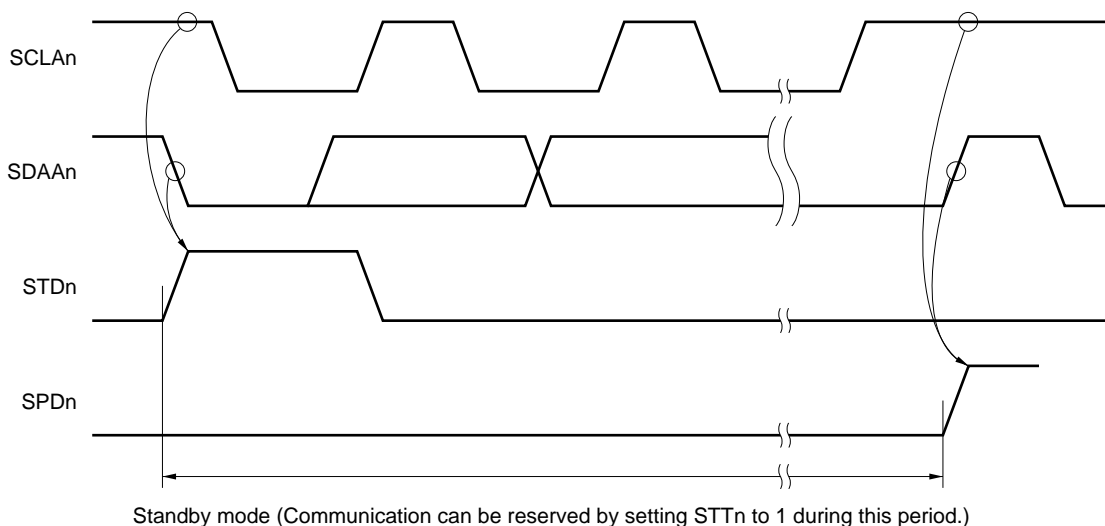
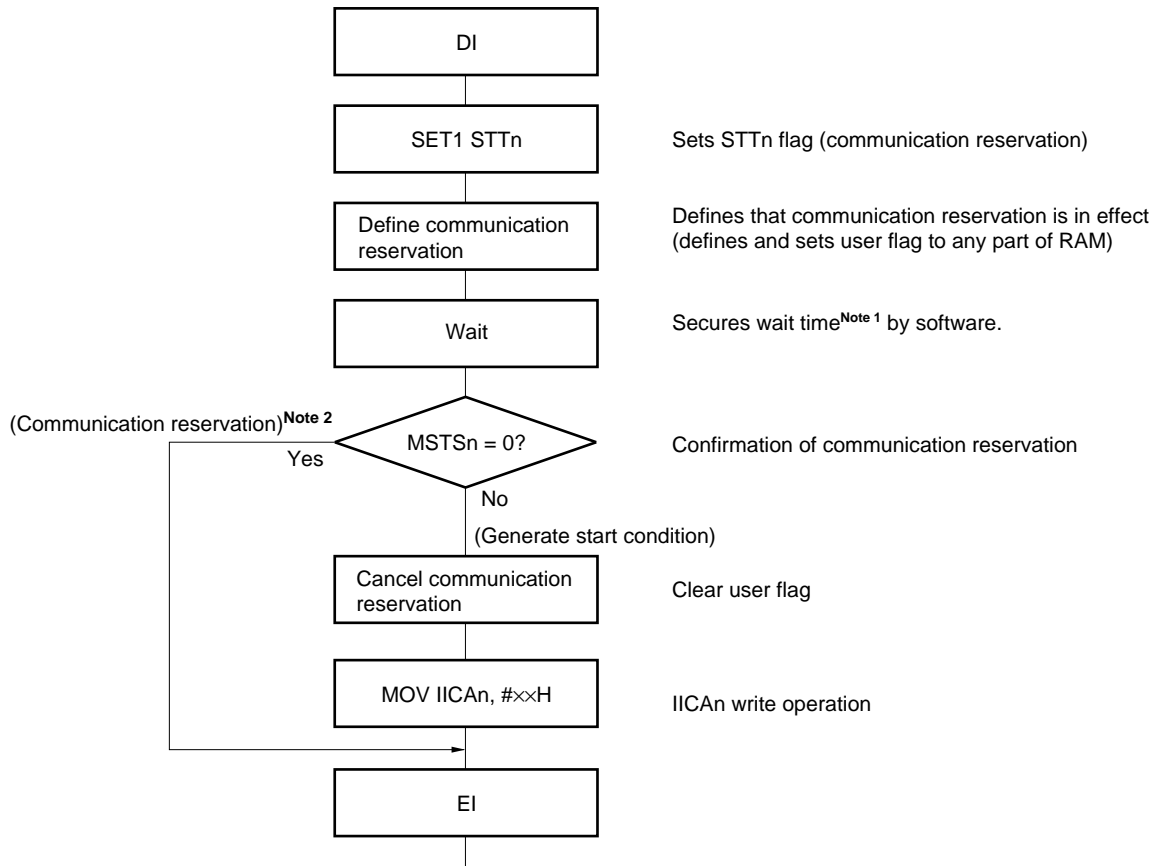


Figure 15-27 shows the communication reservation protocol.

- Remark** n = 0

Figure 15-27. Communication Reservation Protocol



- <R> **Notes** 1. The wait time is calculated as follows.  
 $(IICWL_n \text{ setting value} + IICWH_n \text{ setting value} + 4) / f_{MCK} + t_F \times 2$   
 2. The communication reservation operation executes a write to the IICA shift register n (IICAn) when a stop condition interrupt request occurs.

- Remarks** 1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)  
 MSTSn: Bit 7 of IICA status register n (IICSn)  
 IICAn: IICA shift register n  
 IICWLn: IICA low-level width setting register n  
 IICWHn: IICA high-level width setting register n  
 tF: SDAAn and SCLAn signal falling times  
 fMCK: IICA operation clock frequency  
 2. n = 0

<R>

**(2) When communication reservation function is disabled (bit 0 (IICRSVn) of IICA flag register n (IICFn) = 1)**

When bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LRELn) of the IICCTLn0 register to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCFn (bit 7 of the IICFn register). It takes up to 5 clocks of  $f_{MCK}$  until the STCFn bit is set to 1 after setting STTn = 1. Therefore, secure the time by software.

**Remark** n = 0

### 15.5.15 Cautions

(1) When  $STCENn = 0$

Immediately after I<sup>2</sup>C operation is enabled ( $IICEn = 1$ ), the bus communication status ( $IICBSYn = 1$ ) 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 n1 ( $IICCTLn1$ ).
- <2> Set bit 7 ( $IICEn$ ) of IICA control register n0 ( $IICCTLn0$ ) to 1.
- <3> Set bit 0 ( $SPTn$ ) of the  $IICCTLn0$  register to 1.

(2) When  $STCENn = 1$

Immediately after I<sup>2</sup>C operation is enabled ( $IICEn = 1$ ), the bus released status ( $IICBSYn = 0$ ) is recognized regardless of the actual bus status. To generate the first start condition ( $STTn = 1$ ), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I<sup>2</sup>C communications are already in progress

If I<sup>2</sup>C operation is enabled and the device participates in communication already in progress when the  $SDAAn$  pin is low and the  $SCLAn$  pin is high, the macro of I<sup>2</sup>C recognizes that the  $SDAAn$  pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, ACK is returned, but this interferes with other I<sup>2</sup>C communications. To avoid this, start I<sup>2</sup>C in the following sequence.

- <1> Clear bit 4 ( $SPIEn$ ) of the  $IICCTLn0$  register to 0 to disable generation of an interrupt request signal ( $INTIICAn$ ) when the stop condition is detected.
- <2> Set bit 7 ( $IICEn$ ) of the  $IICCTLn0$  register to 1 to enable the operation of I<sup>2</sup>C.
- <3> Wait for detection of the start condition.
- <4> Set bit 6 ( $LRELn$ ) of the  $IICCTLn0$  register to 1 before ACK is returned (4 to 72 clocks of  $f_{MCK}$  after setting the  $IICEn$  bit to 1), to forcibly disable detection.

(4) Setting the  $STTn$  and  $SPTn$  bits (bits 1 and 0 of the  $IICCTLn0$  register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set the  $SPIEn$  bit (bit 4 of the  $IICCTLn0$  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 n ( $IICAn$ ) 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  $SPIEn$  bit to 1 when the  $MSTS$  bit (bit 7 of the IICA status register n ( $IICSn$ )) is detected by software.

**Remark** n = 0

### 15.5.16 Communication operations

The following shows three operation procedures with the flowchart.

#### (1) Master operation in single master system

The flowchart when using the RL78/L13 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<sup>2</sup>C bus multimaster system, whether the bus is released or used cannot be judged by the I<sup>2</sup>C 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/L13 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/L13 loses 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/L13 is used as the I<sup>2</sup>C bus slave is shown below.

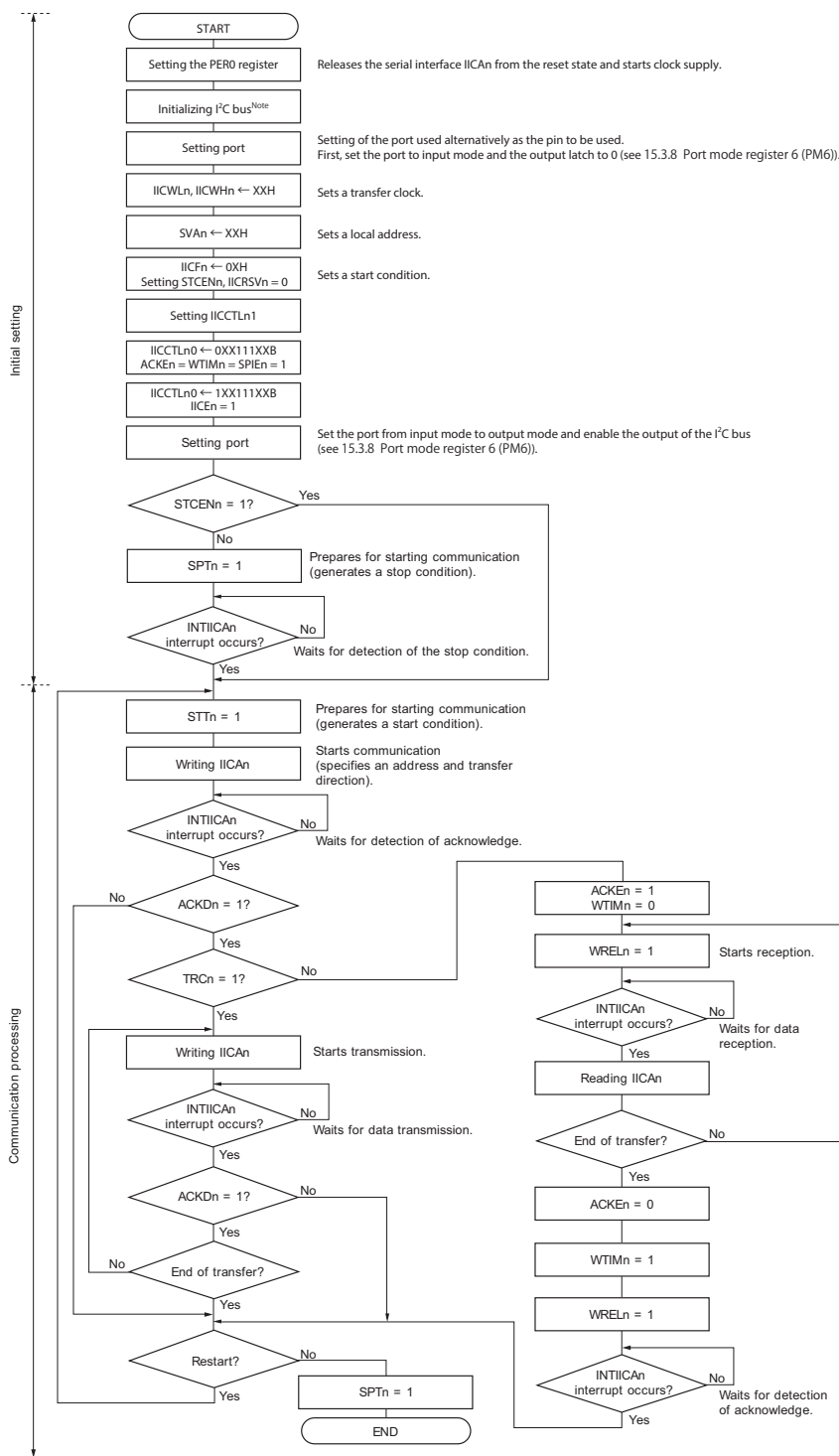
When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICAn interrupt occurrence (communication waiting). When an INTIICAn interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

**Remark** n = 0

(1) Master operation in single-master system

Figure 15-28. Master Operation in Single-Master System



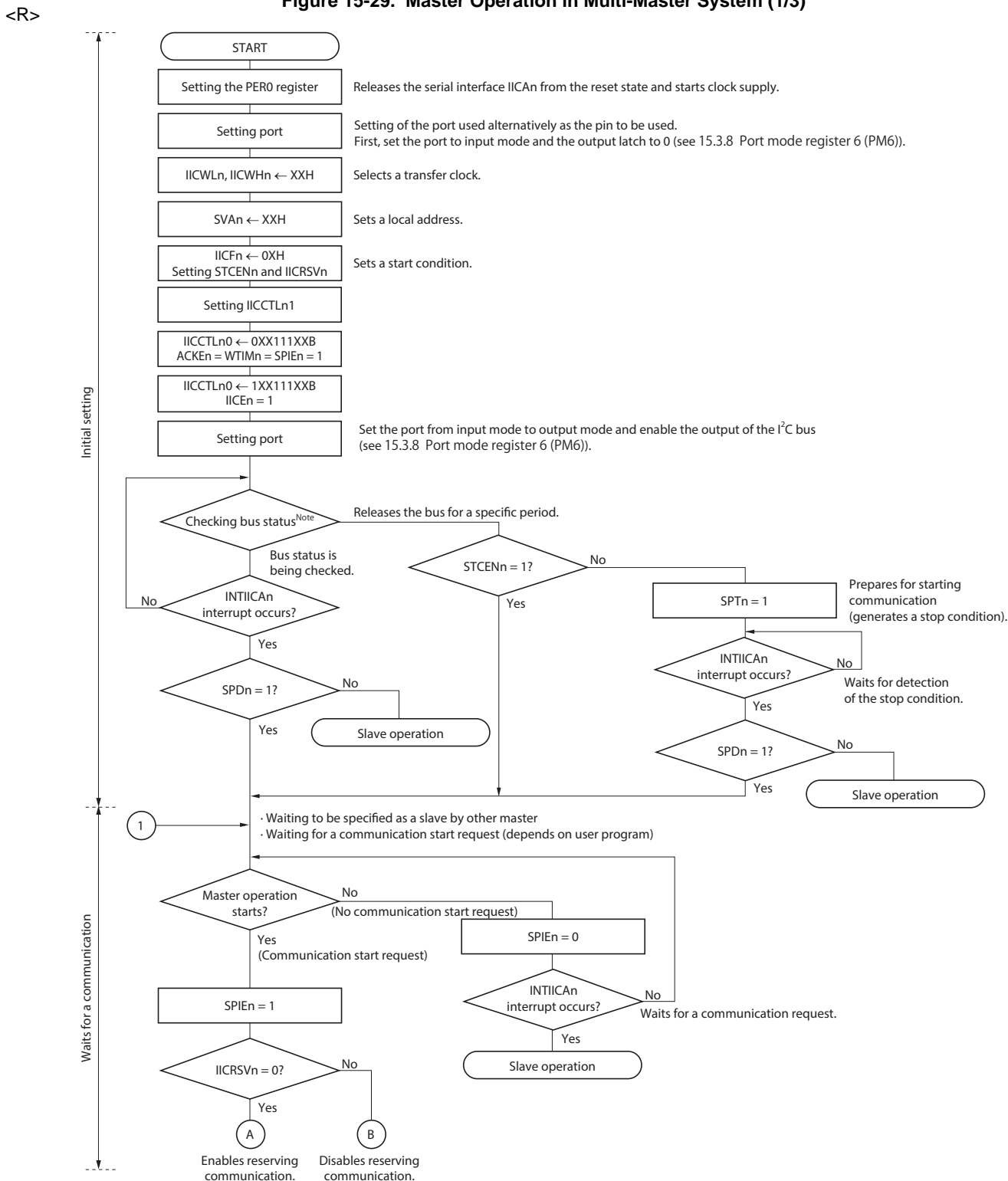
**Note** Release (SCLAn and SDAAn pins = high level) the I<sup>2</sup>C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the SDAAn pin is constantly at high level.

- Remarks 1.** Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.
- 2.** n = 0



(2) Master operation in multi-master system

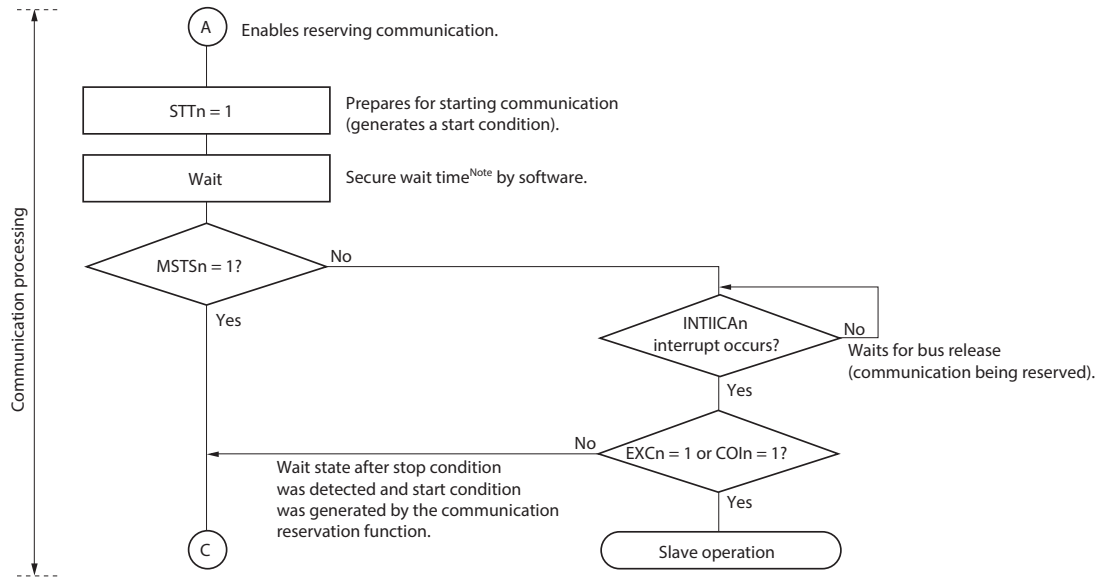
Figure 15-29. Master Operation in Multi-Master System (1/3)



**Note** Confirm that the bus is released (CLDn bit = 1, DADn bit = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I<sup>2</sup>C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

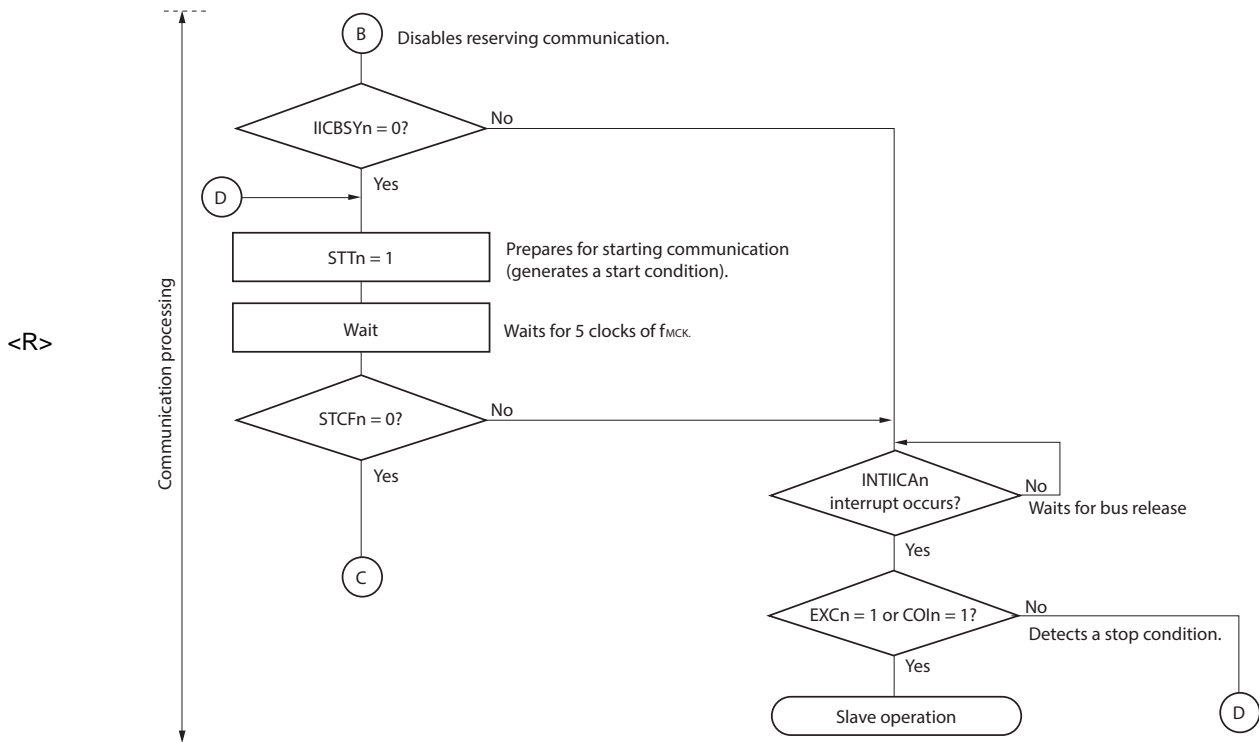
**Remark** n = 0

Figure 15-29. Master Operation in Multi-Master System (2/3)



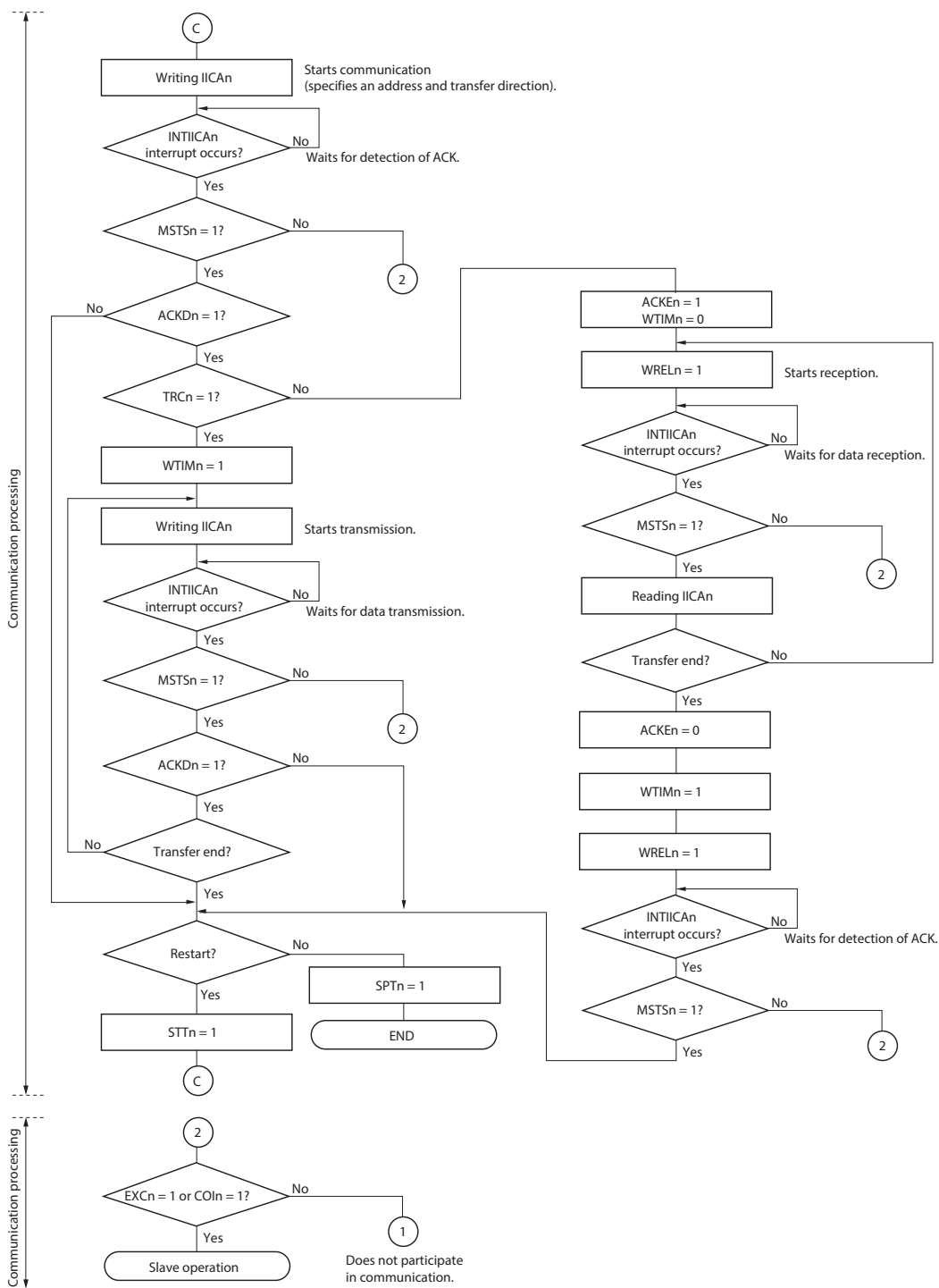
<R> **Note** The wait time is calculated as follows.  

$$(IICWLn \text{ setting value} + IICWHn \text{ setting value} + 4) / f_{MCK} + t_F \times 2$$



- Remarks 1.** IICWLn: IICA low-level width setting register n  
 IICWHn: IICA high-level width setting register n  
 t<sub>F</sub>: SDAAn and SCLAn signal falling times  
 f<sub>MCK</sub>: IICA operation clock frequency
- 2.** n = 0

Figure 15-29. Master Operation in Multi-Master System (3/3)



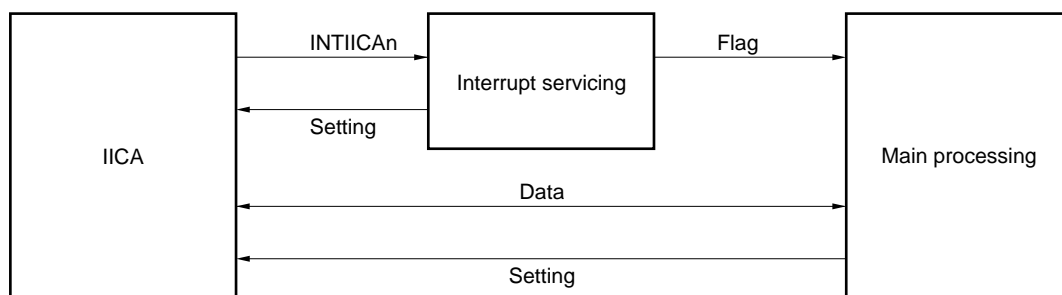
- Remarks**
1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.
  2. To use the device as a master in a multi-master system, read the MSTSn bit each time interrupt INTIICAn has occurred to check the arbitration result.
  3. To use the device as a slave in a multi-master system, check the status by using the IICA status register n (IICSn) and IICA flag register n (IICFn) each time interrupt INTIICAn has occurred, and determine the processing to be performed next.
  4. n = 0

**(3) Slave operation**

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICAn 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 INTIICAn 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 INTIICAn.

**<1> Communication mode flag**

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of ACK from master, address mismatch)

**<2> Ready flag**

This flag indicates that data communication is enabled. Its function is the same as the INTIICAn 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 TRCn bit.

**Remark** n = 0

The main processing of the slave operation is explained next.

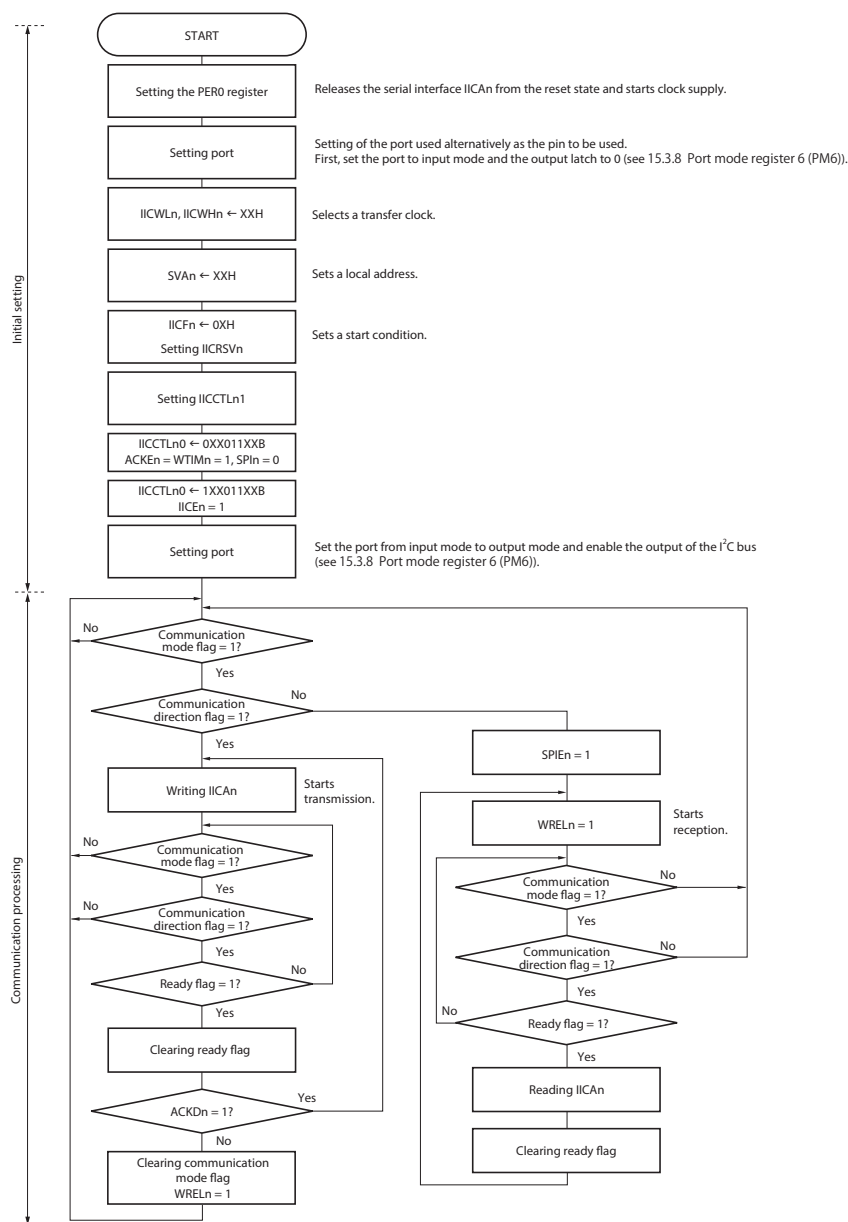
Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns ACK. If ACK is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, ACK is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Figure 15-30. Slave Operation Flowchart (1)

<R>



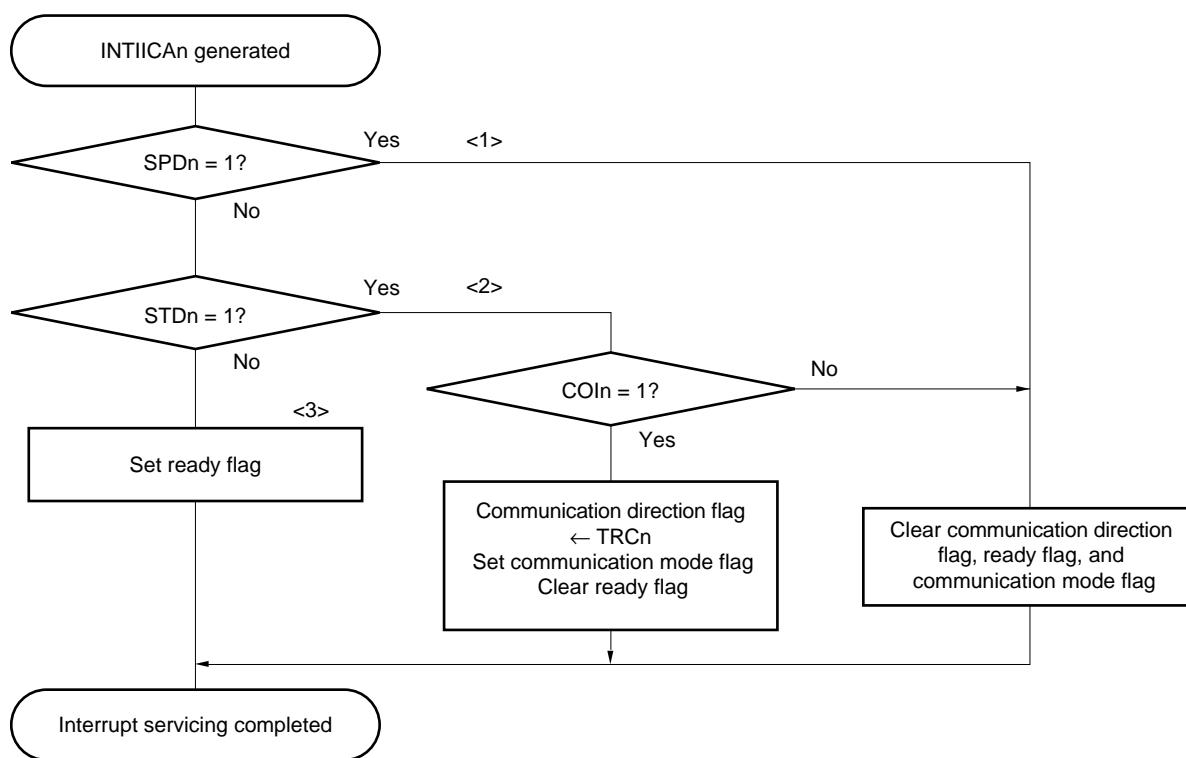
- Remarks 1. Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.
- 2. n = 0

An example of the processing procedure of the slave with the INTIICAn interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICAn interrupt checks the status, and the following operations are performed.

- <1> Communication is stopped if the stop condition is issued.
- <2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).
- <3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I<sup>2</sup>C bus remaining in the wait state.

**Remark** <1> to <3> above correspond to <1> to <3> in Figure 15-31 Slave Operation Flowchart (2).

**Figure 15-31. Slave Operation Flowchart (2)**



**Remark** n = 0

**15.5.17 Timing of I<sup>2</sup>C interrupt request (INTIICAn) occurrence**

The timing of transmitting or receiving data and generation of interrupt request signal INTIICAn, and the value of the IICA status register n (IICSn) when the INTIICAn signal is generated are shown below.

- Remarks 1.**
- ST: Start condition
  - AD6 to AD0: Address
  - R/W: Transfer direction specification
  - ACK: Acknowledge
  - D7 to D0: Data
  - SP: Stop condition
- 2.** n = 0

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIMn = 0

ST	AD6 to AD0	R $\bar{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1		▲2		▲3 ▲4	△5

SPTn = 1  
↓

▲1: IICSn = 1000x110B  
 ▲2: IICSn = 1000x000B  
 ▲3: IICSn = 1000x000B (Sets the WTIMn bit to 1)<sup>Note</sup>  
 ▲4: IICSn = 1000xx00B (Sets the SPTn bit to 1)<sup>Note</sup>  
 △5: IICSn = 00000001B

**Note** To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

**Remark** ▲: Always generated  
 △: Generated only when SPIEn = 1  
 x: Don't care

(ii) When WTIMn = 1

ST	AD6 to AD0	R $\bar{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1		▲2		▲3	△4

SPTn = 1  
↓

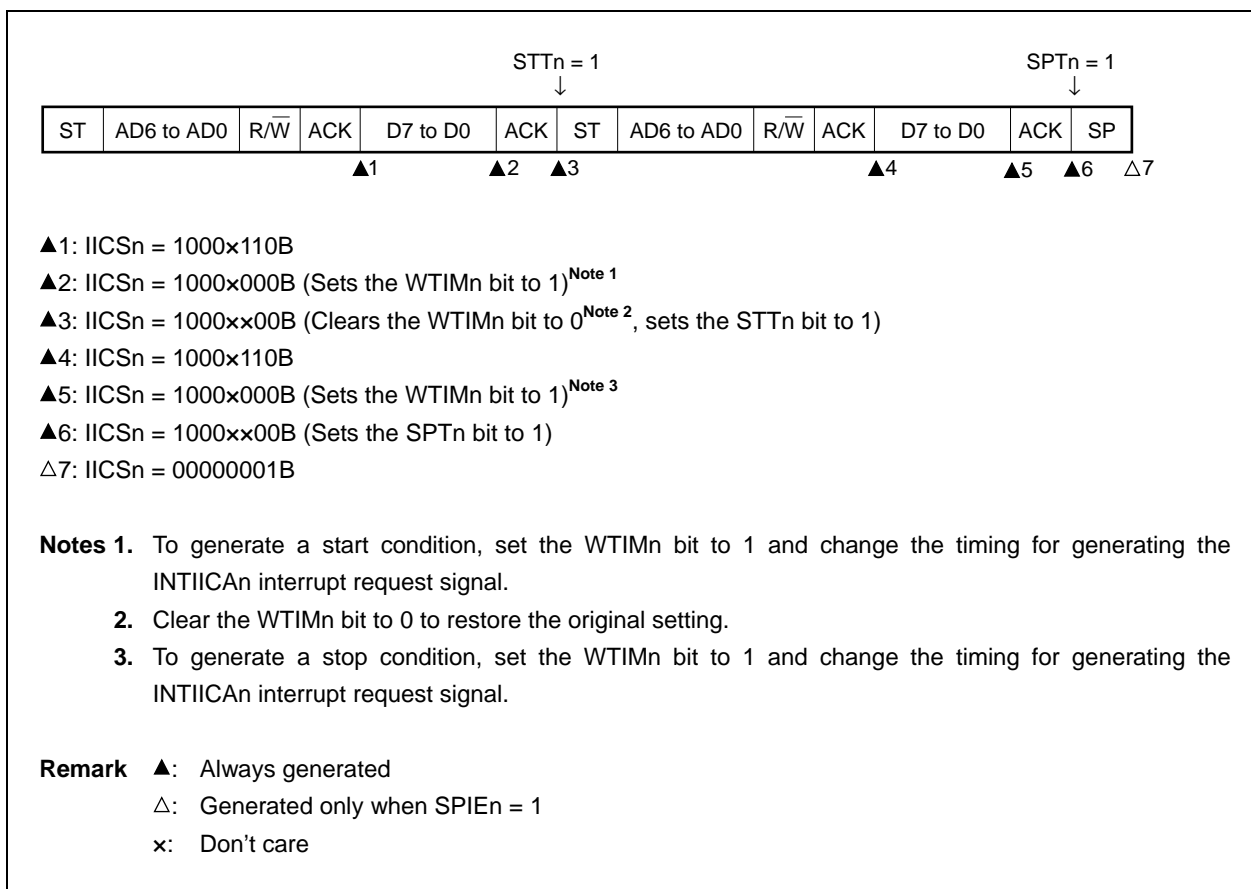
▲1: IICSn = 1000x110B  
 ▲2: IICSn = 1000x100B  
 ▲3: IICSn = 1000xx00B (Sets the SPTn bit to 1)  
 △4: IICSn = 00000001B

**Remark** ▲: Always generated  
 △: Generated only when SPIEn = 1  
 x: Don't care

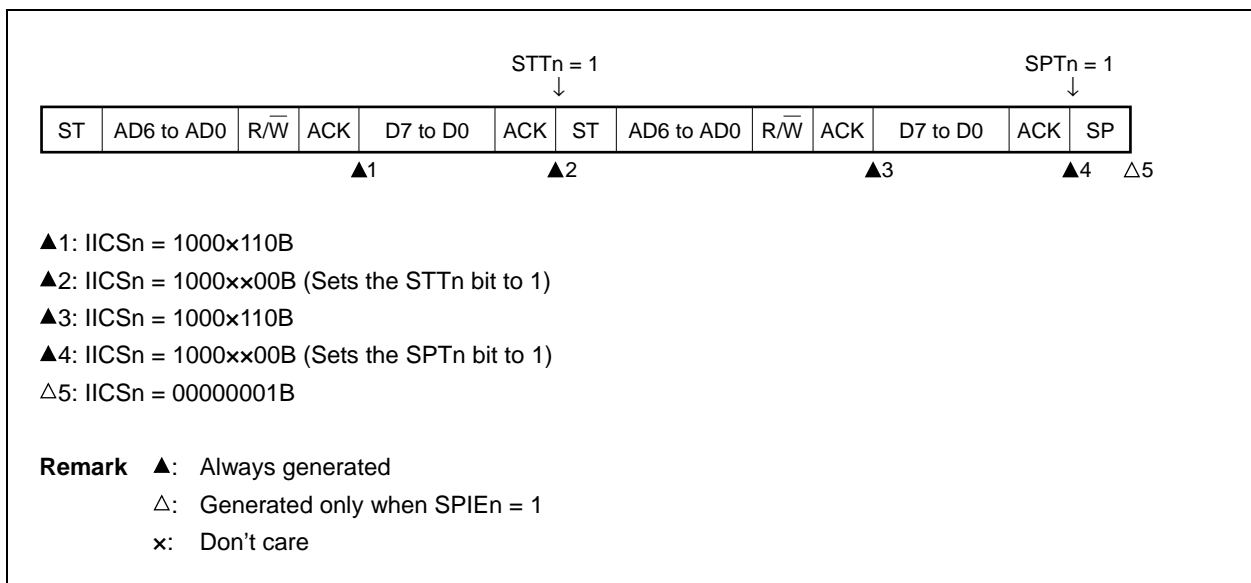
**Remark** n = 0

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIMn = 0



(ii) When WTIMn = 1



**Remark** n = 0



(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIMn = 0

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1		▲2		▲3 ▲4	△5

SPTn = 1  
↓

▲1: IICSn = 1010x110B  
 ▲2: IICSn = 1010x000B  
 ▲3: IICSn = 1010x000B (Sets the WTIMn bit to 1)<sup>Note</sup>  
 ▲4: IICSn = 1010xx00B (Sets the SPTn bit to 1)  
 △5: IICSn = 00000001B

**Note** To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

**Remark** ▲: Always generated  
 △: Generated only when SPIEn = 1  
 x: Don't care

(ii) When WTIMn = 1

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1		▲2		▲3	△4

SPTn = 1  
↓

▲1: IICSn = 1010x110B  
 ▲2: IICSn = 1010x100B  
 ▲3: IICSn = 1010xx00B (Sets the SPTn bit to 1)  
 △4: IICSn = 00001001B

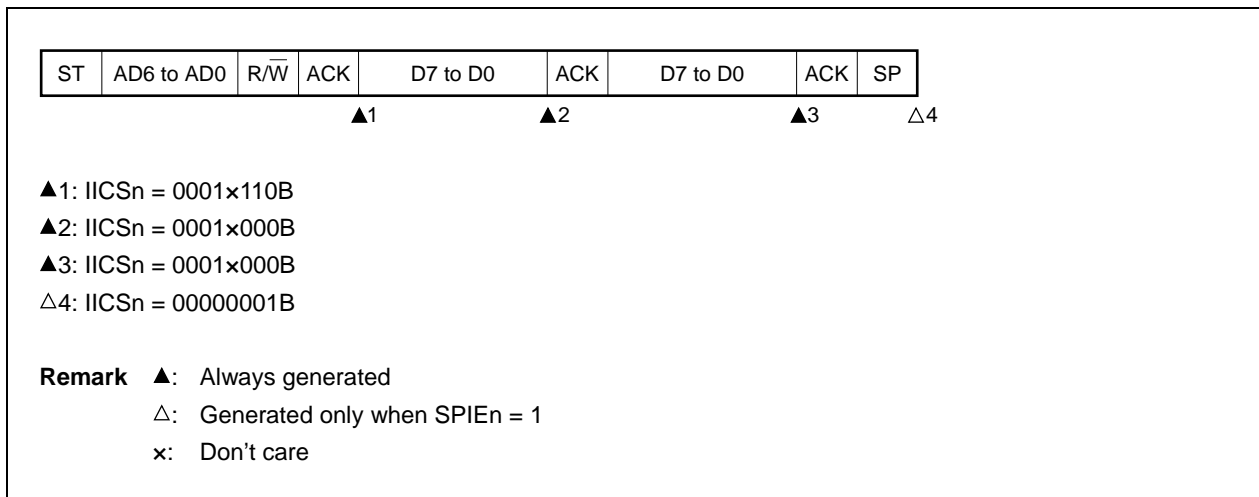
**Remark** ▲: Always generated  
 △: Generated only when SPIEn = 1  
 x: Don't care

**Remark** n = 0

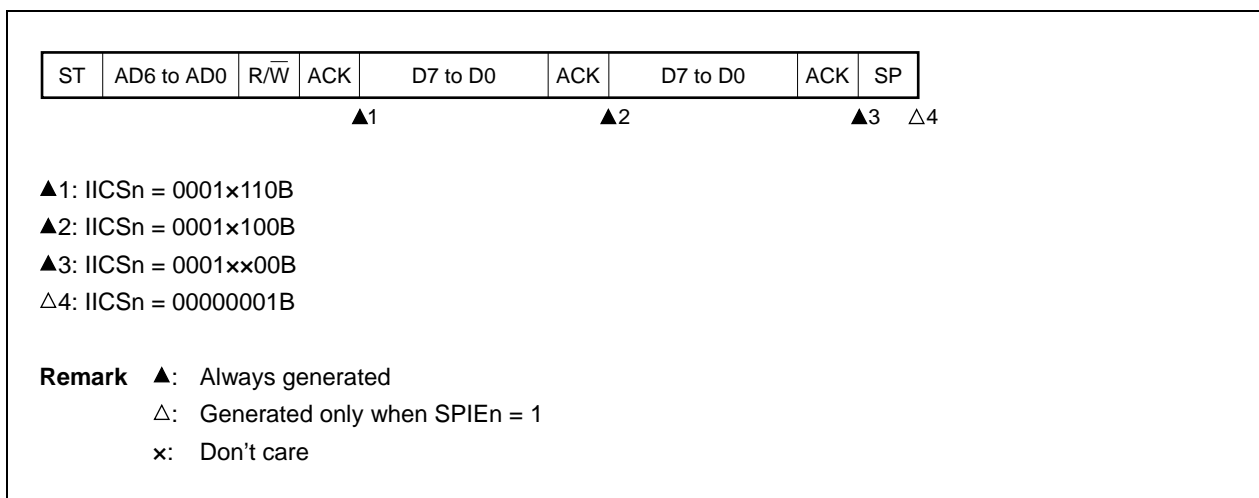
**(2) Slave device operation (slave address data reception)**

**(a) Start ~ Address ~ Data ~ Data ~ Stop**

**(i) When WTIMn = 0**



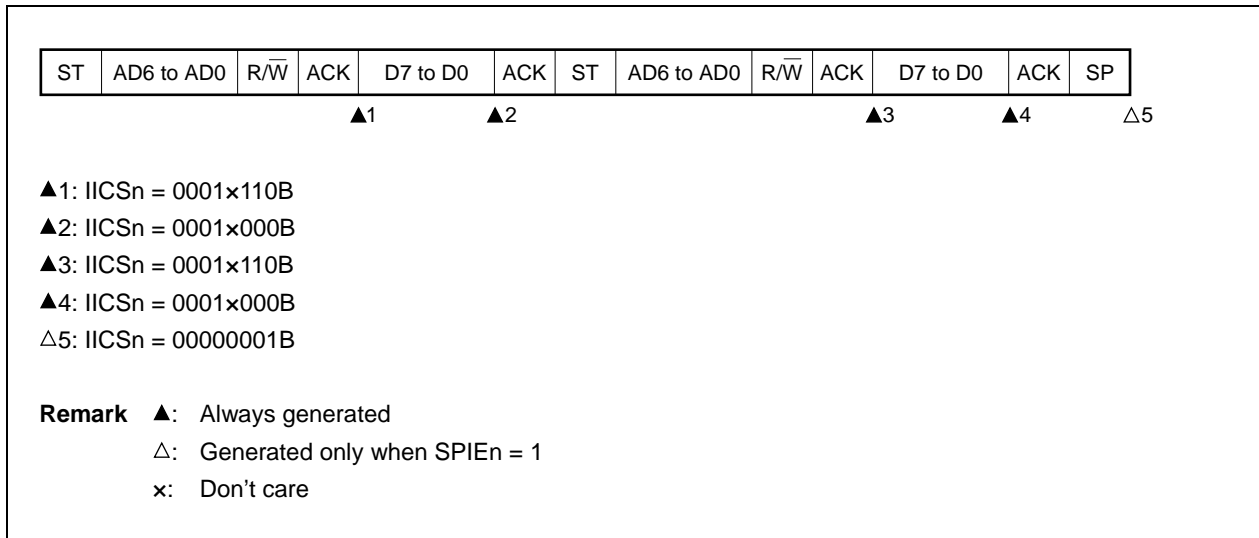
**(ii) When WTIMn = 1**



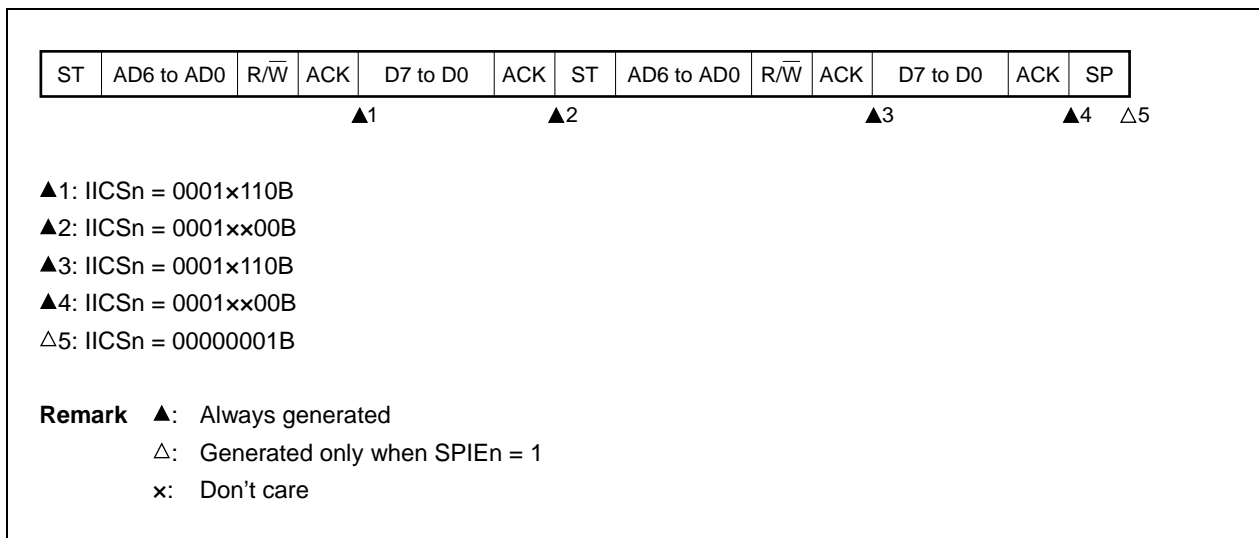
**Remark** n = 0

**(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop**

**(i) When WTIMn = 0 (after restart, matches with SVAn)**



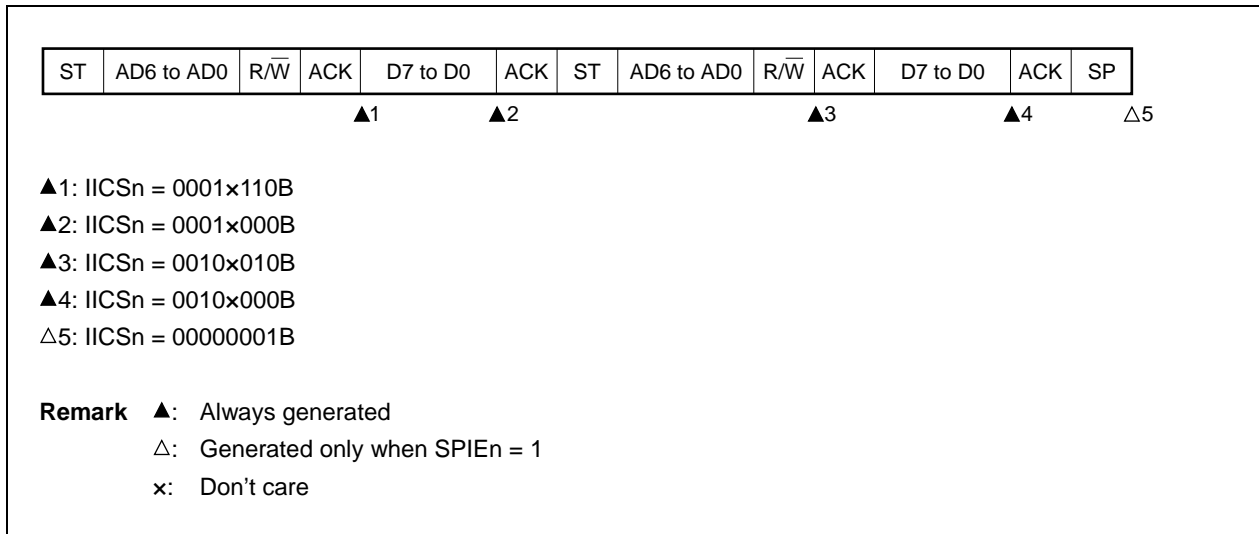
**(ii) When WTIMn = 1 (after restart, matches with SVAn)**



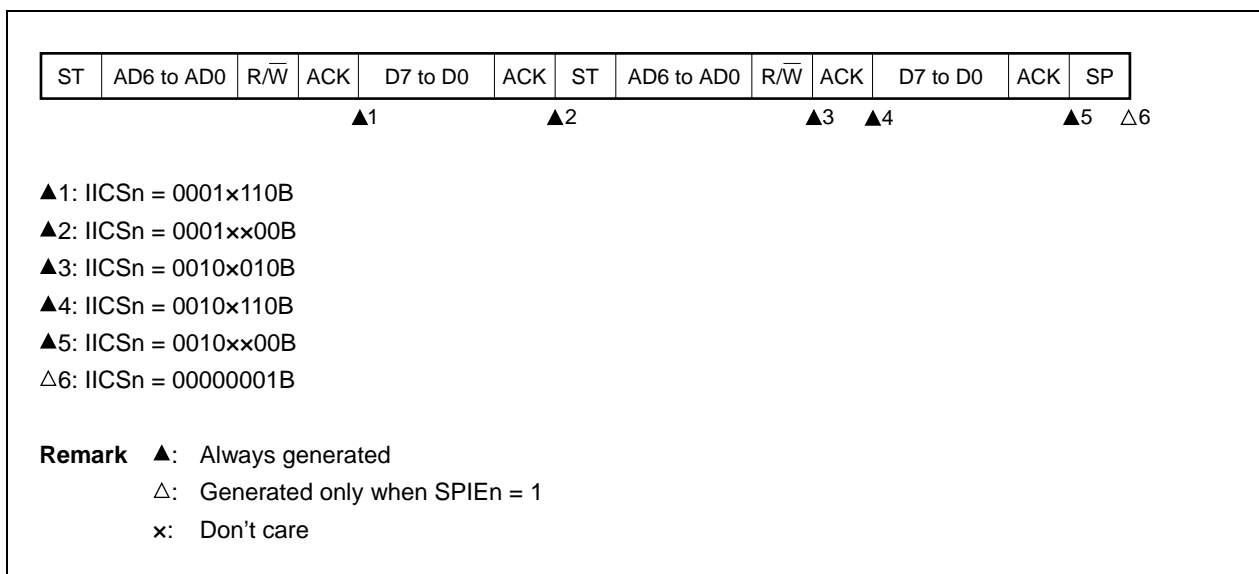
**Remark** n = 0

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= extension code))



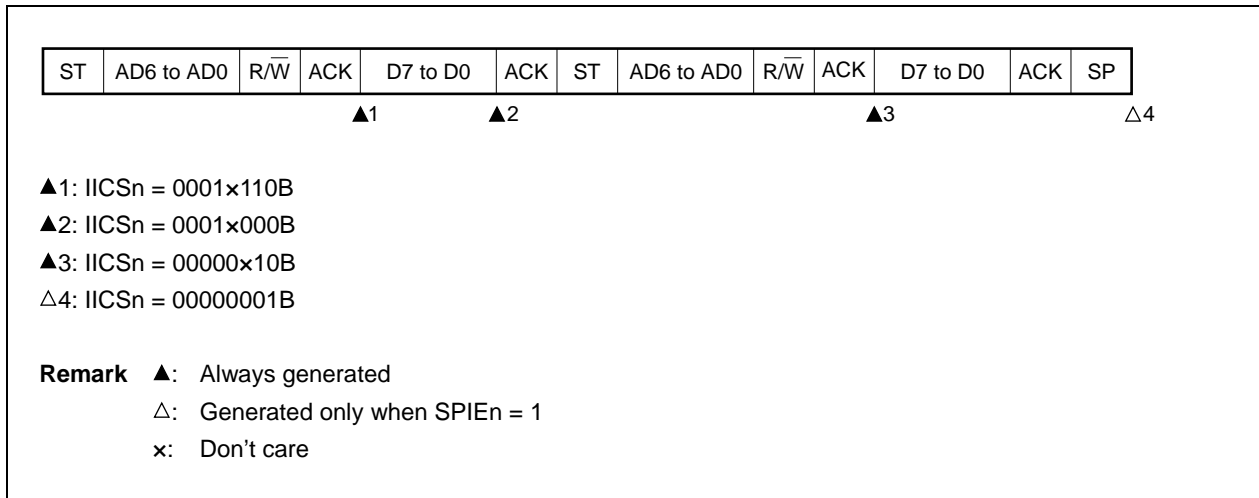
(ii) When WTIMn = 1 (after restart, does not match address (= extension code))



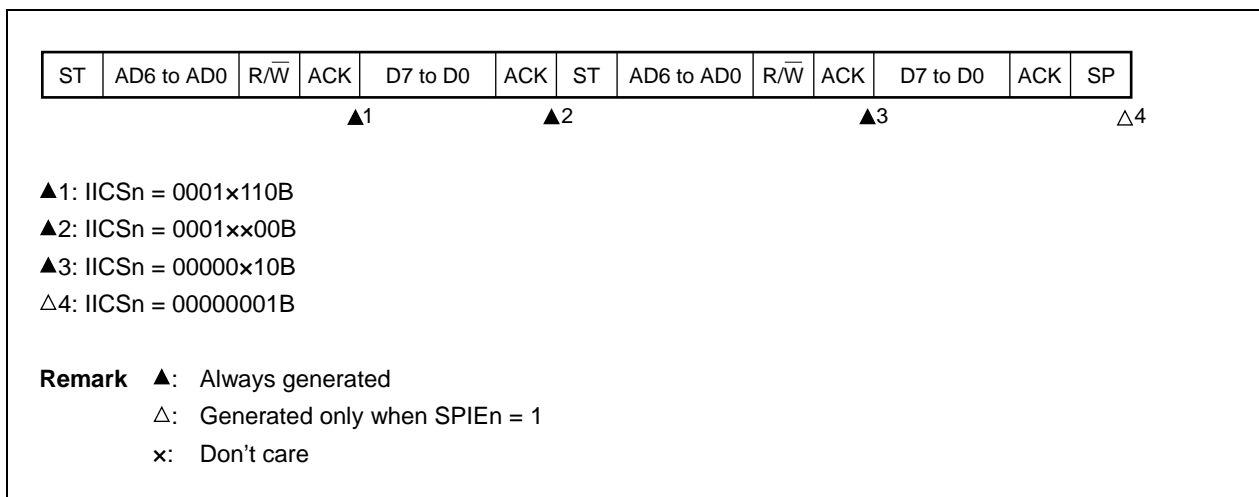
**Remark** n = 0

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= not extension code))



(ii) When WTIMn = 1 (after restart, does not match address (= not extension code))



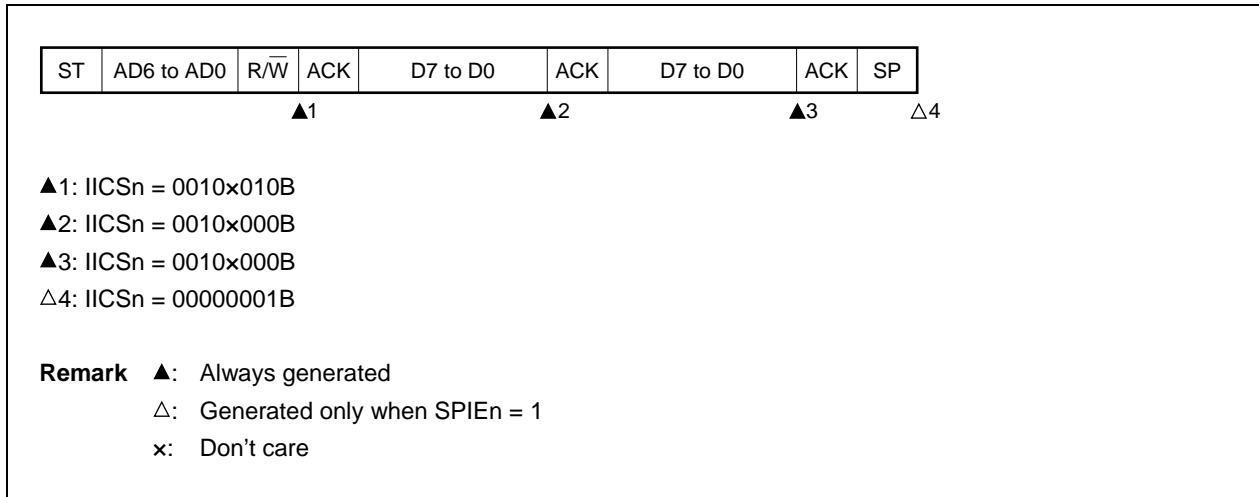
**Remark** n = 0

**(3) Slave device operation (when receiving extension code)**

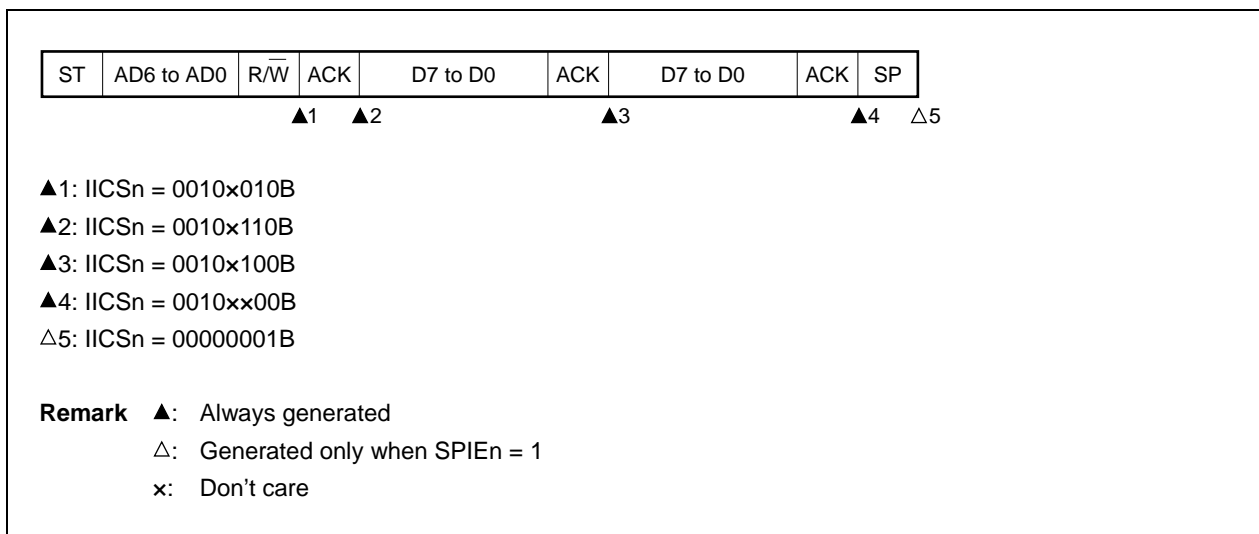
The device is always participating in communication when it receives an extension code.

**(a) Start ~ Code ~ Data ~ Data ~ Stop**

**(i) When WTIMn = 0**



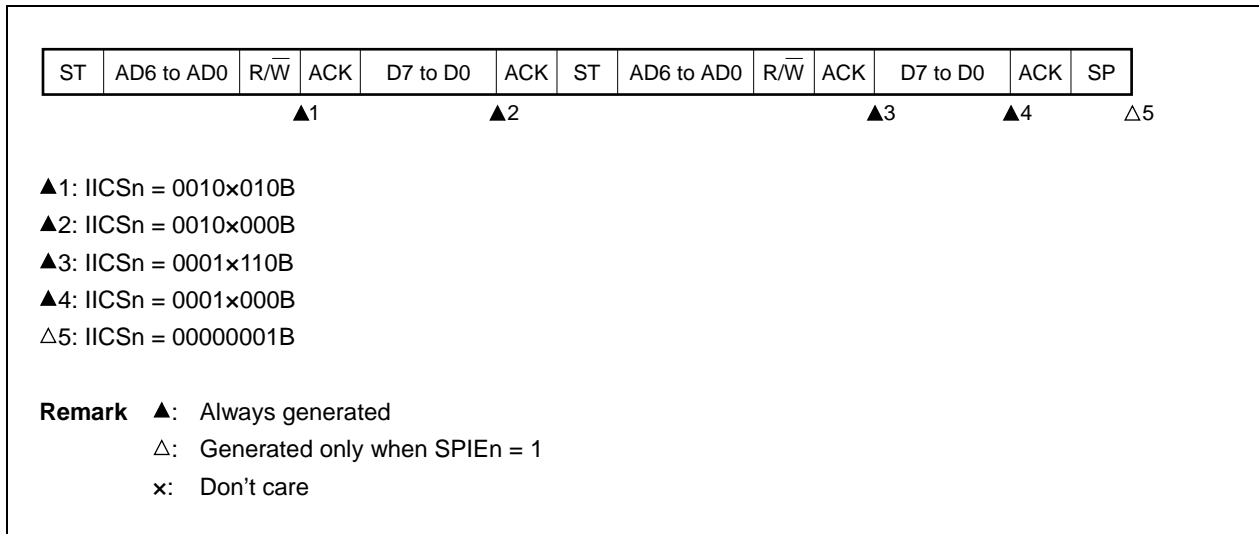
**(ii) When WTIMn = 1**



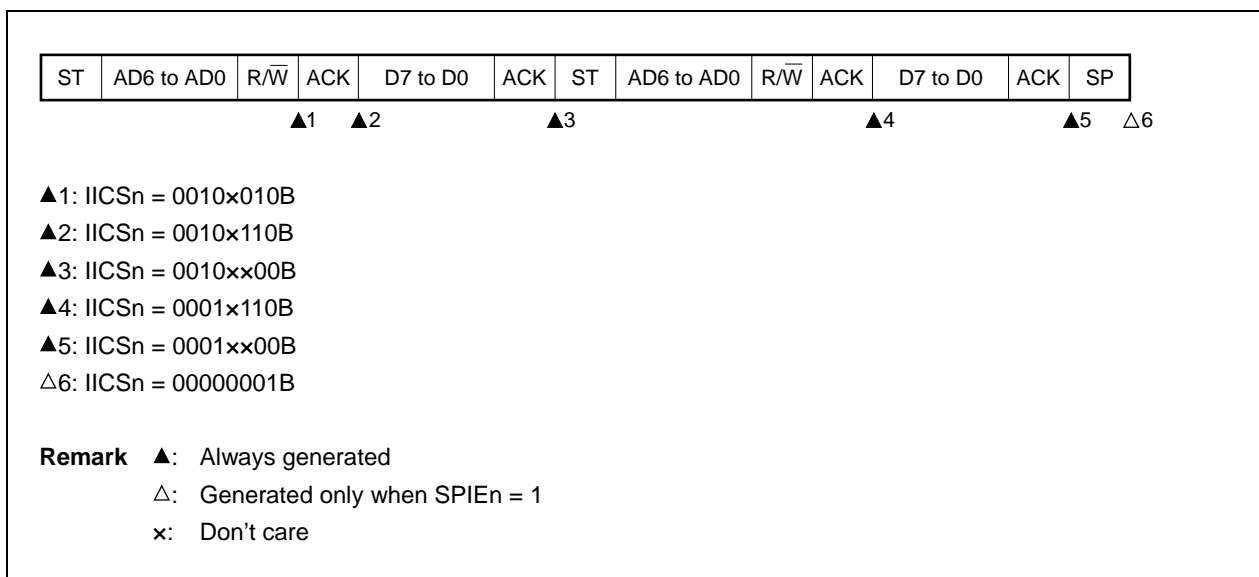
**Remark** n = 0

**(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop**

**(i) When WTIMn = 0 (after restart, matches SVAn)**



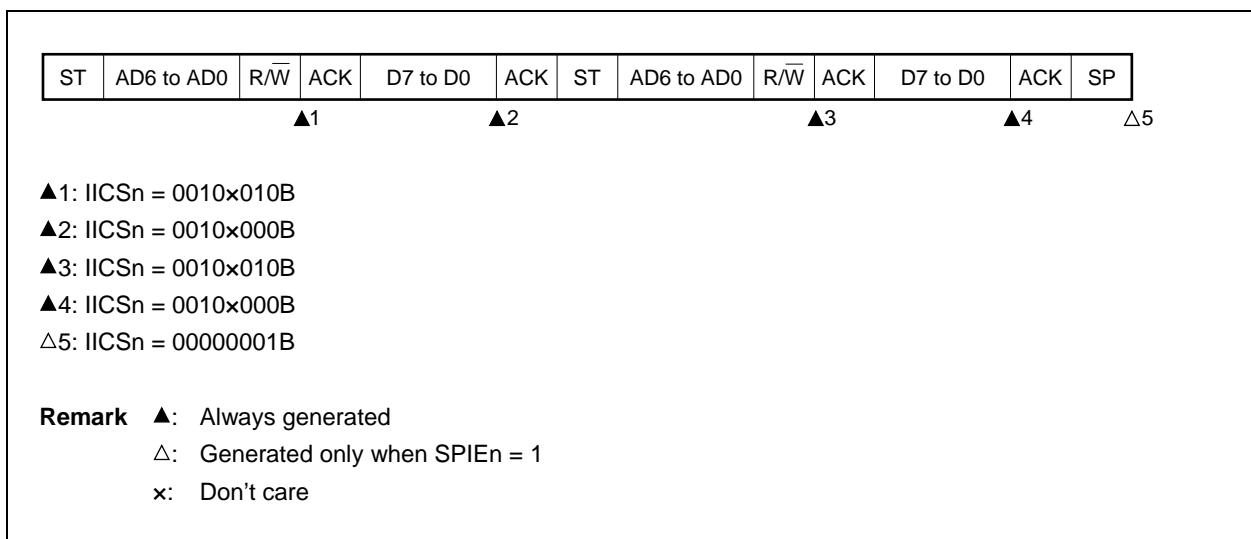
**(ii) When WTIMn = 1 (after restart, matches SVAn)**



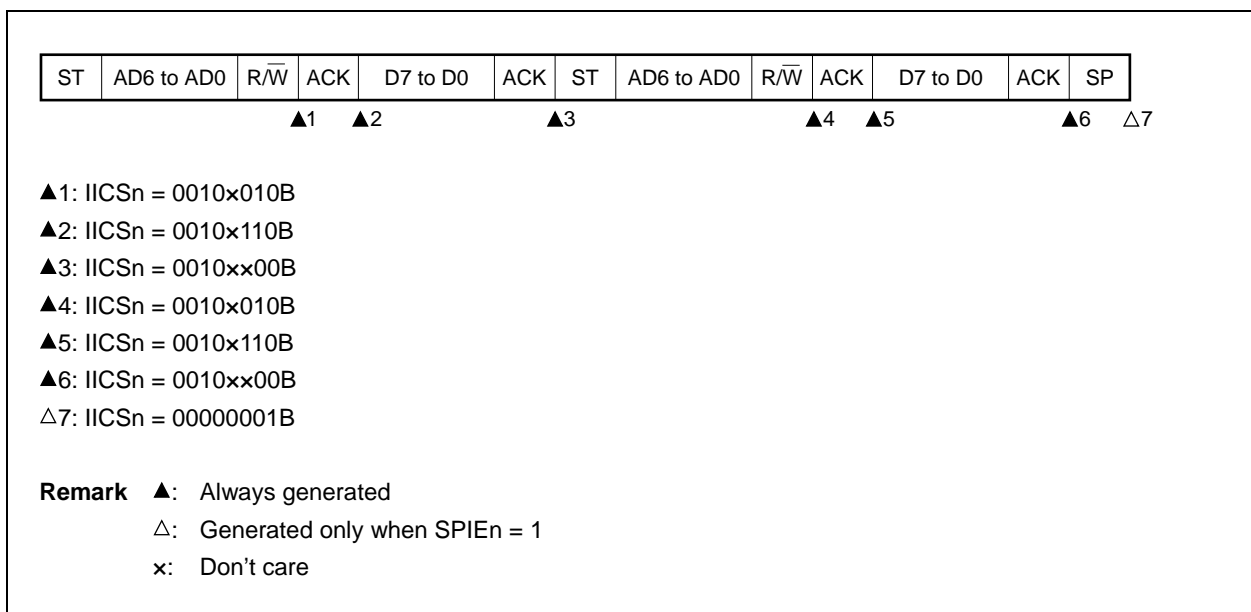
**Remark** n = 0

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, extension code reception)



(ii) When WTIMn = 1 (after restart, extension code reception)

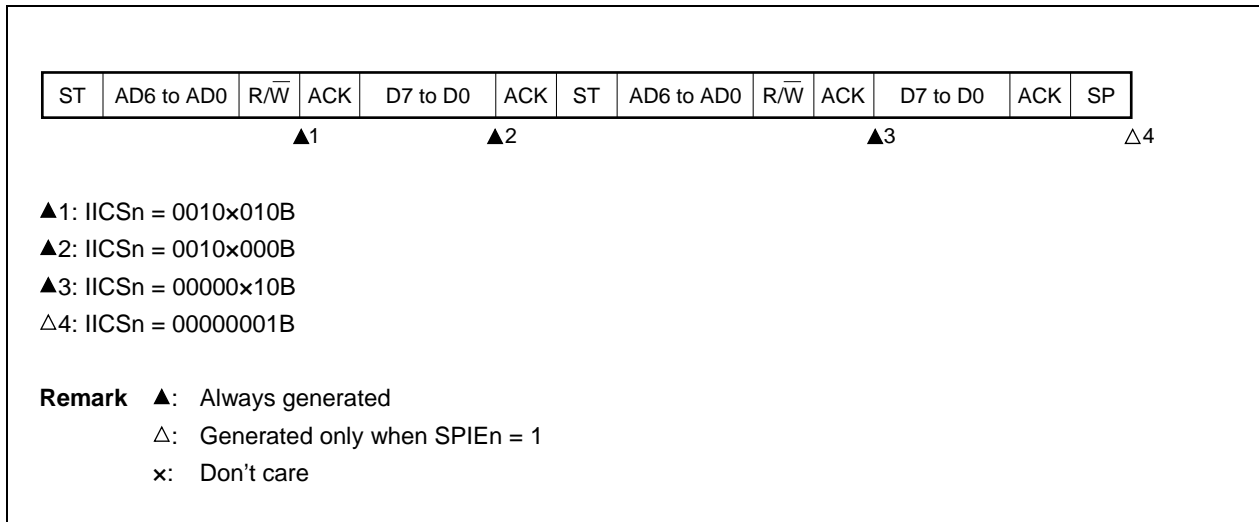


**Remark** n = 0

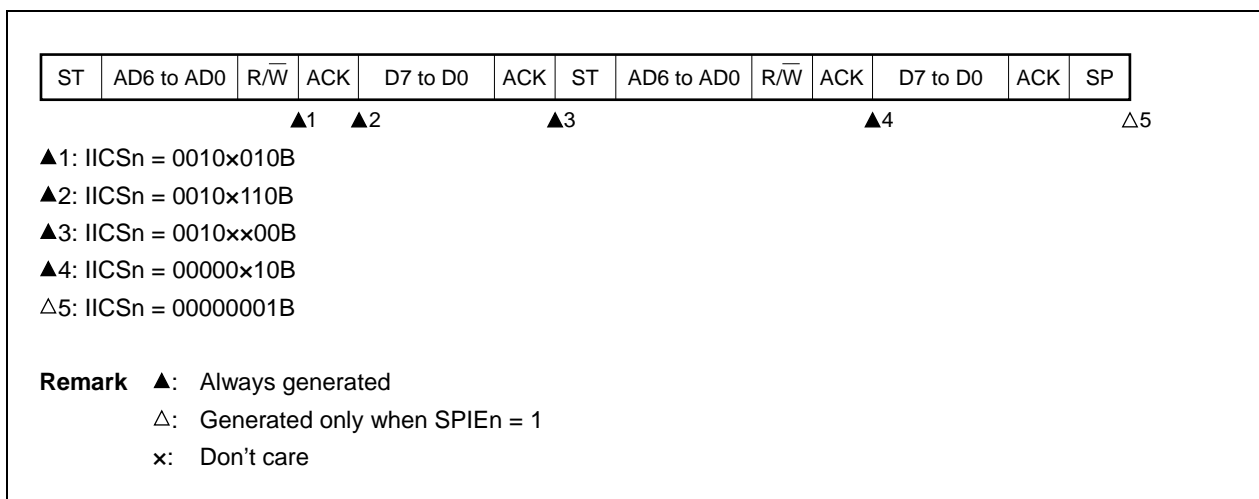


(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When  $WTIMn = 0$  (after restart, does not match address (= not extension code))



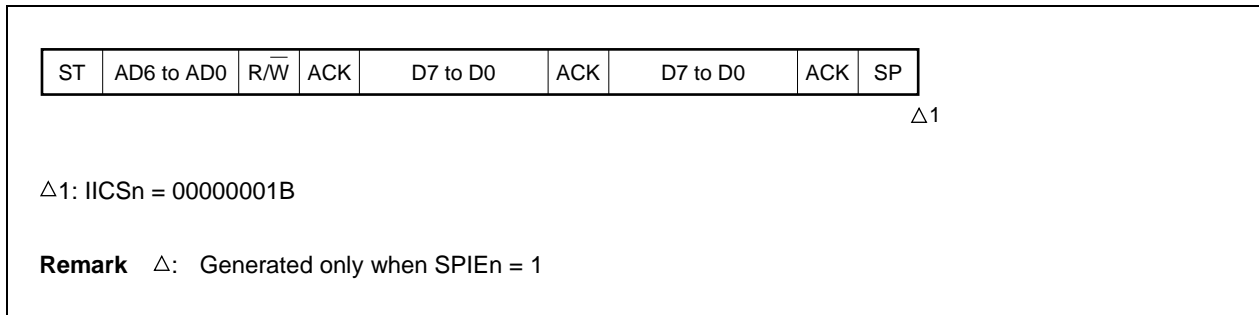
(ii) When  $WTIMn = 1$  (after restart, does not match address (= not extension code))



**Remark** n = 0

**(4) Operation without communication**

**(a) Start ~ Code ~ Data ~ Data ~ Stop**

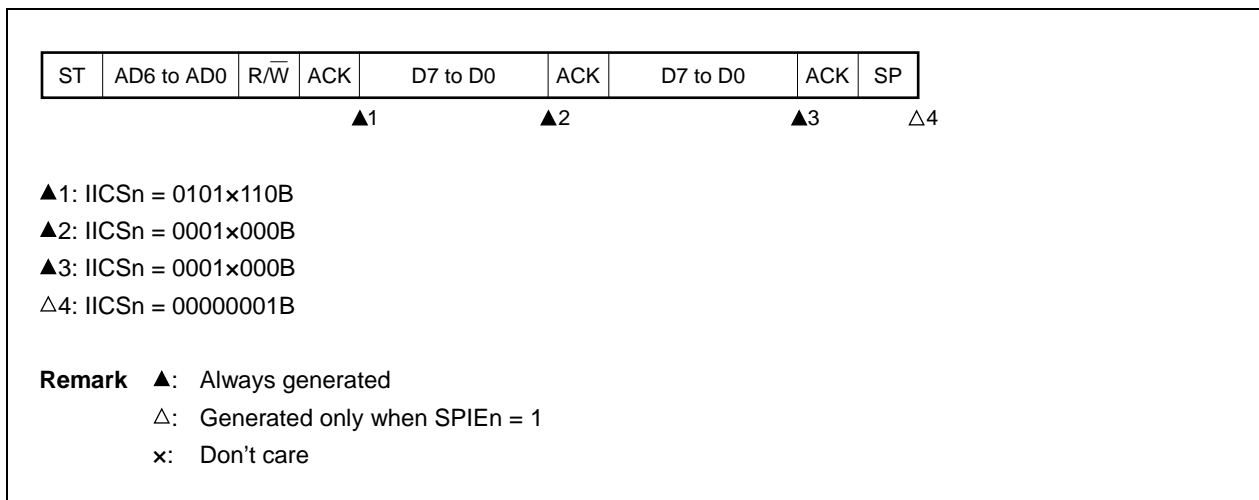


**(5) Arbitration loss operation (operation as slave after arbitration loss)**

When the device is used as a master in a multi-master system, read the MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

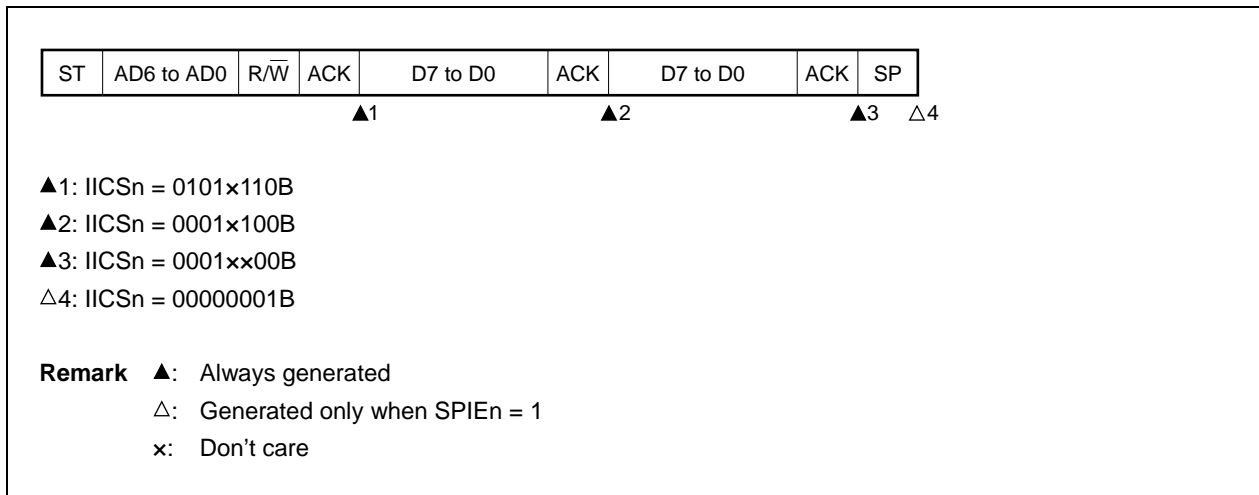
**(a) When arbitration loss occurs during transmission of slave address data**

**(i) When WTIMn = 0**



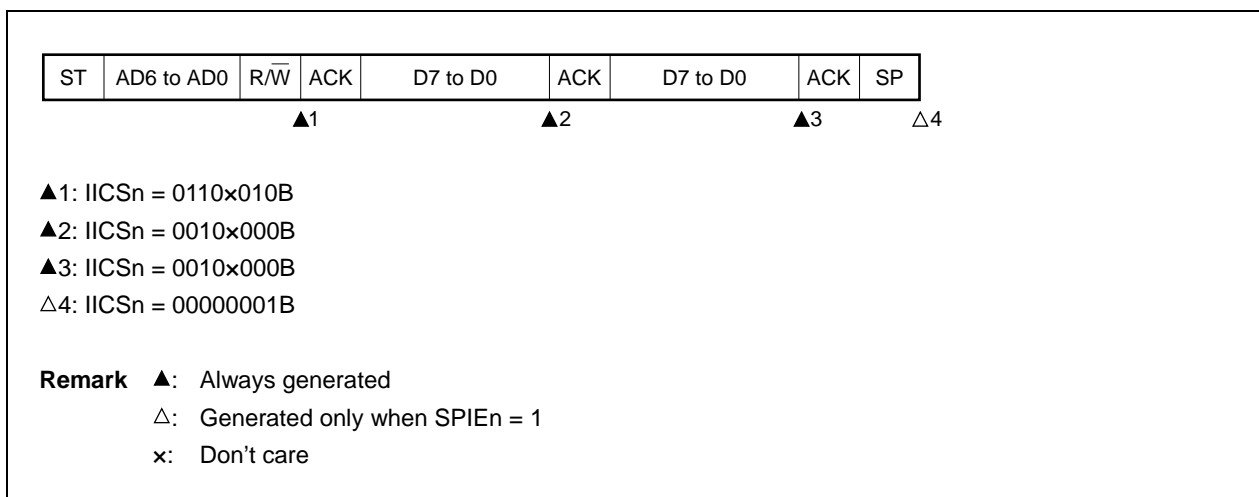
**Remark** n = 0

(ii) When WTIMn = 1



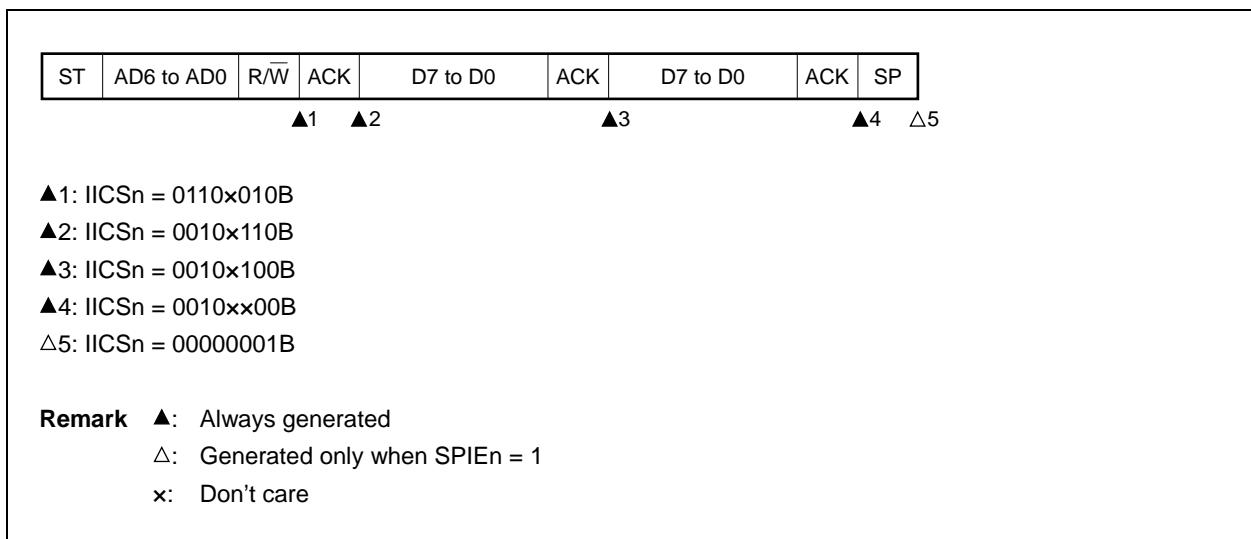
(b) When arbitration loss occurs during transmission of extension code

(i) When WTIMn = 0



**Remark** n = 0

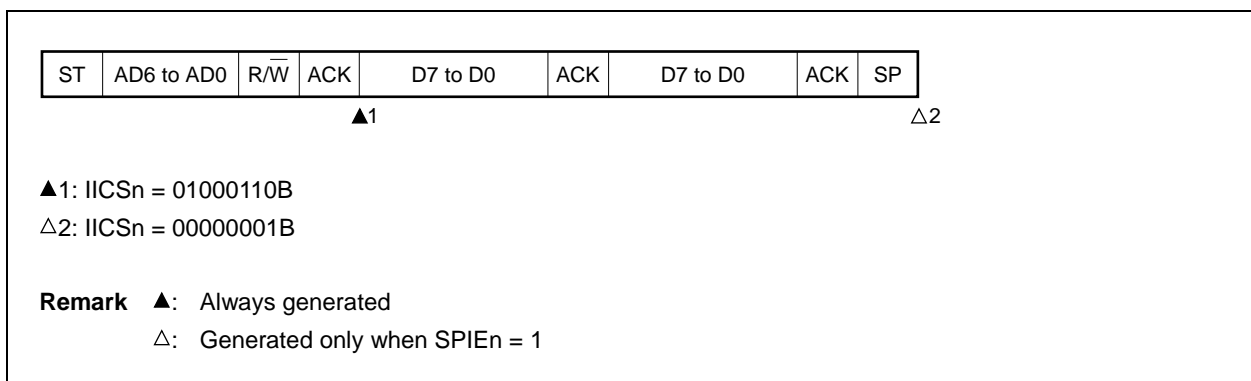
(ii) When WTIMn = 1



(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 MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data (when WTIMn = 1)



**Remark** n = 0

**(b) When arbitration loss occurs during transmission of extension code**

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1				△2	

▲1: IICSn = 0110x010B  
 Sets LRELn = 1 by software  
 △2: IICSn = 00000001B

**Remark** ▲: Always generated  
 △: Generated only when SPIEn = 1  
 x: Don't care

**(c) When arbitration loss occurs during transmission of data**

**(i) When WTIMn = 0**

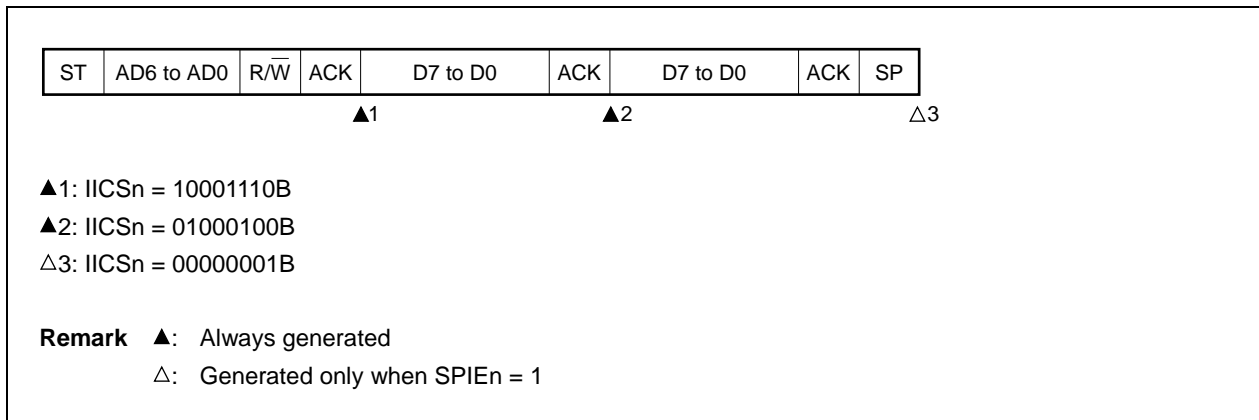
ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1		▲2			△3

▲1: IICSn = 10001110B  
 ▲2: IICSn = 01000000B  
 △3: IICSn = 00000001B

**Remark** ▲: Always generated  
 △: Generated only when SPIEn = 1

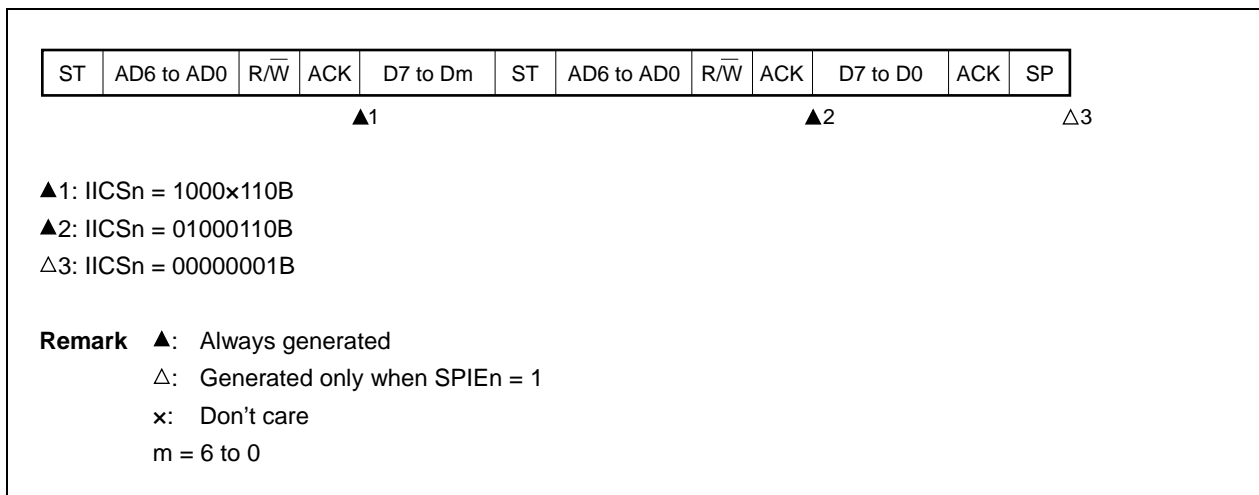
**Remark** n = 0

(ii) When WTIMn = 1



(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatched with SVAn)



**Remark** n = 0

(ii) Extension code

ST	AD6 to AD0	R/W	ACK	D7 to Dm	ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	SP
			▲1				▲2				△3

▲1: IICSn = 1000x110B  
 ▲2: IICSn = 01100010B  
 Sets LRELn = 1 by software  
 △3: IICSn = 00000001B

**Remark** ▲: Always generated  
 △: Generated only when SPIEn = 1  
 x: Don't care  
 m = 6 to 0

(e) When loss occurs due to stop condition during data transfer

ST	AD6 to AD0	R/W	ACK	D7 to Dm	SP
			▲1		△2

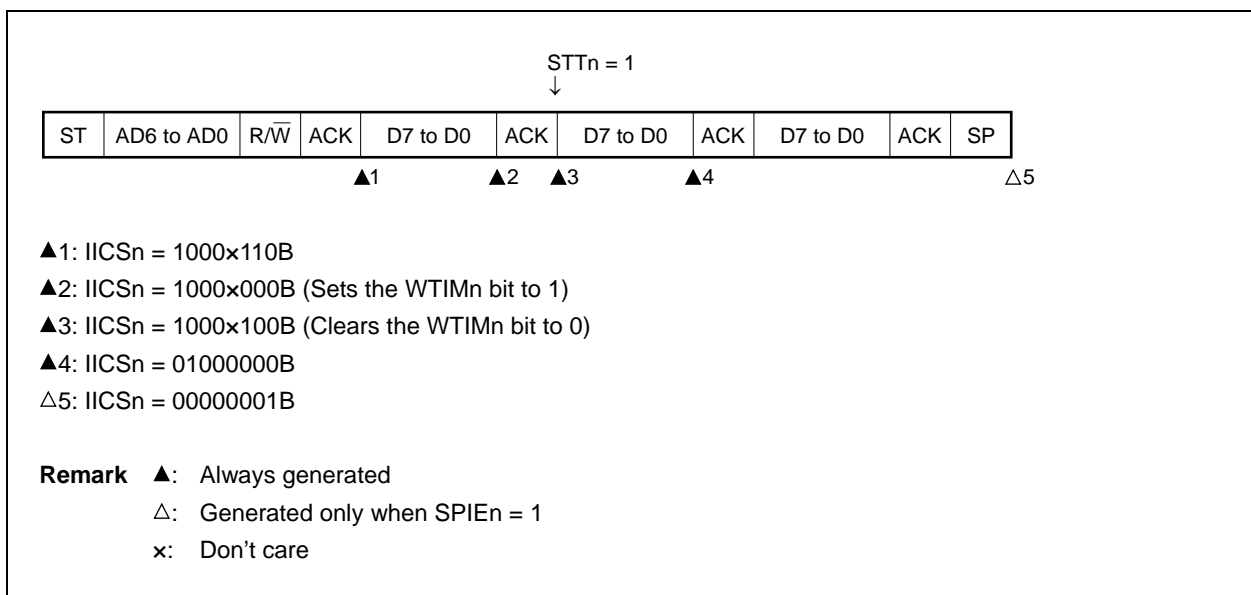
▲1: IICSn = 10000110B  
 △2: IICSn = 01000001B

**Remark** ▲: Always generated  
 △: Generated only when SPIEn = 1  
 x: Don't care  
 m = 6 to 0

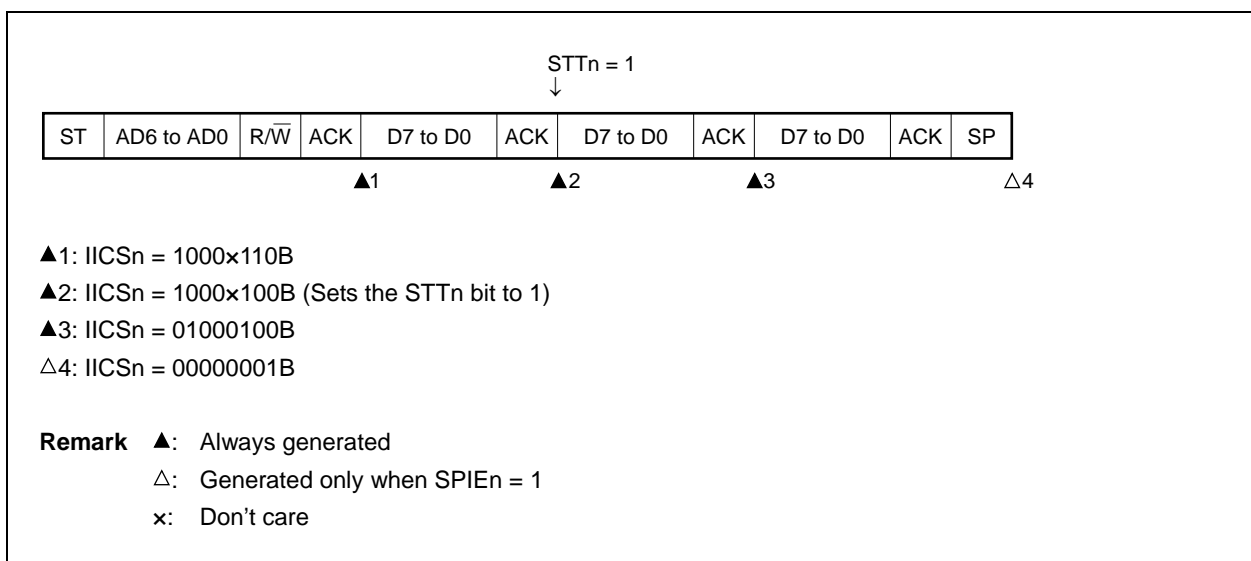
**Remark** n = 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

(i) When  $WTIMn = 0$



(ii) When  $WTIMn = 1$

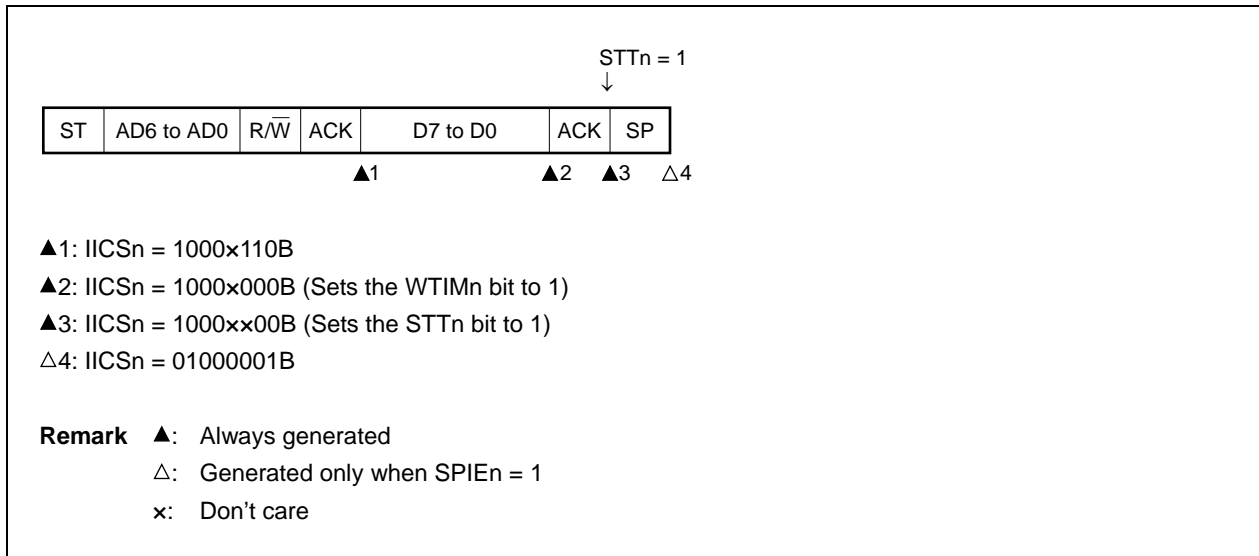


**Remark** n = 0

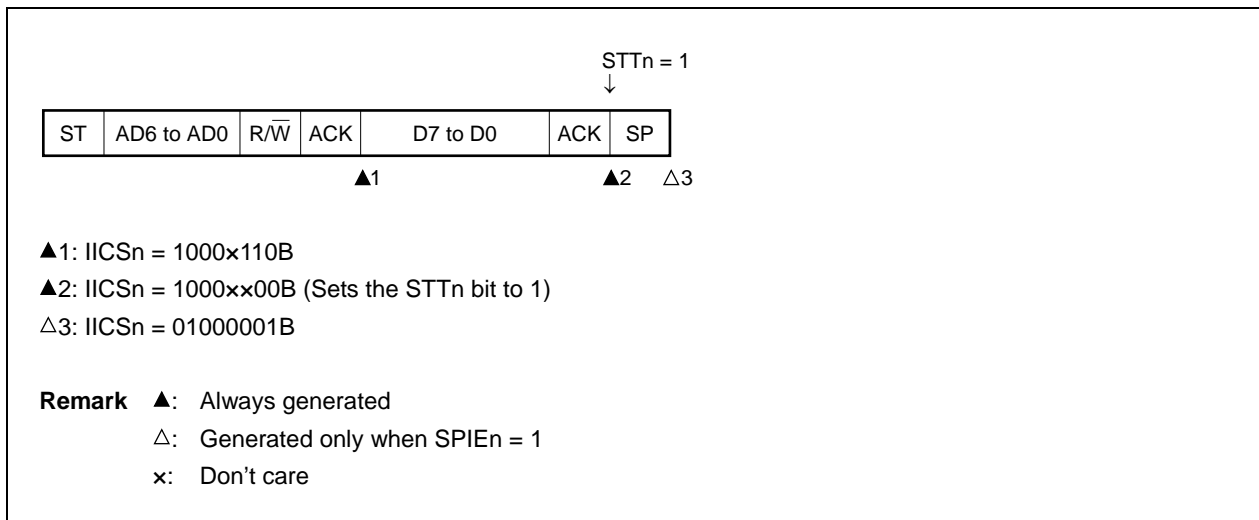


(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

(i) When  $WTIMn = 0$



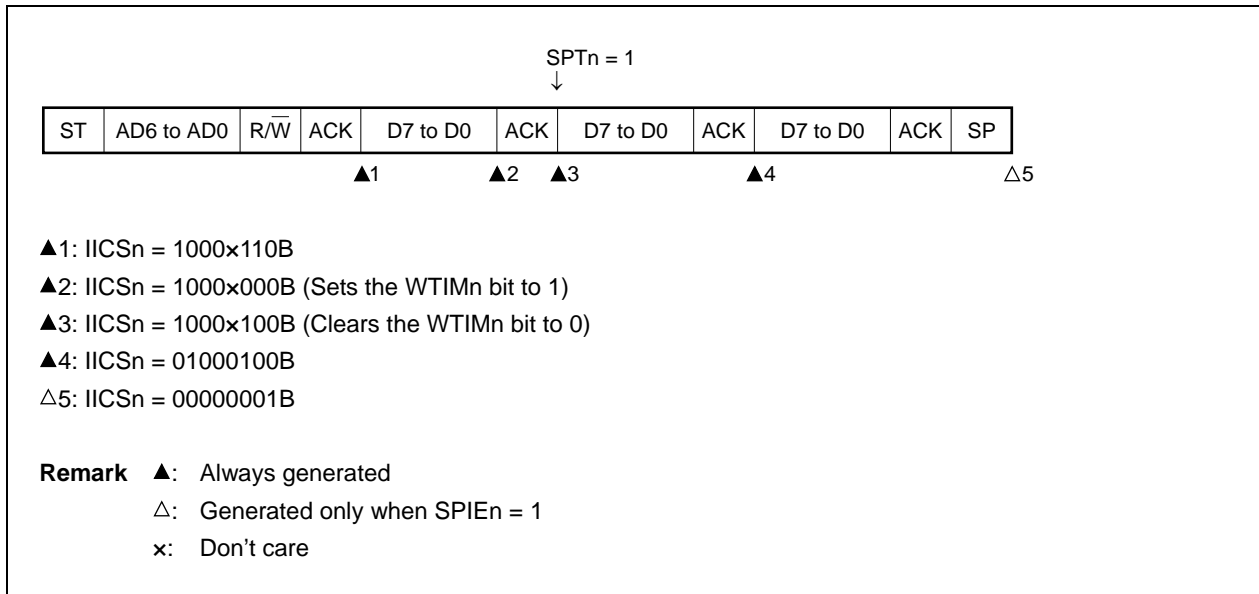
(ii) When  $WTIMn = 1$



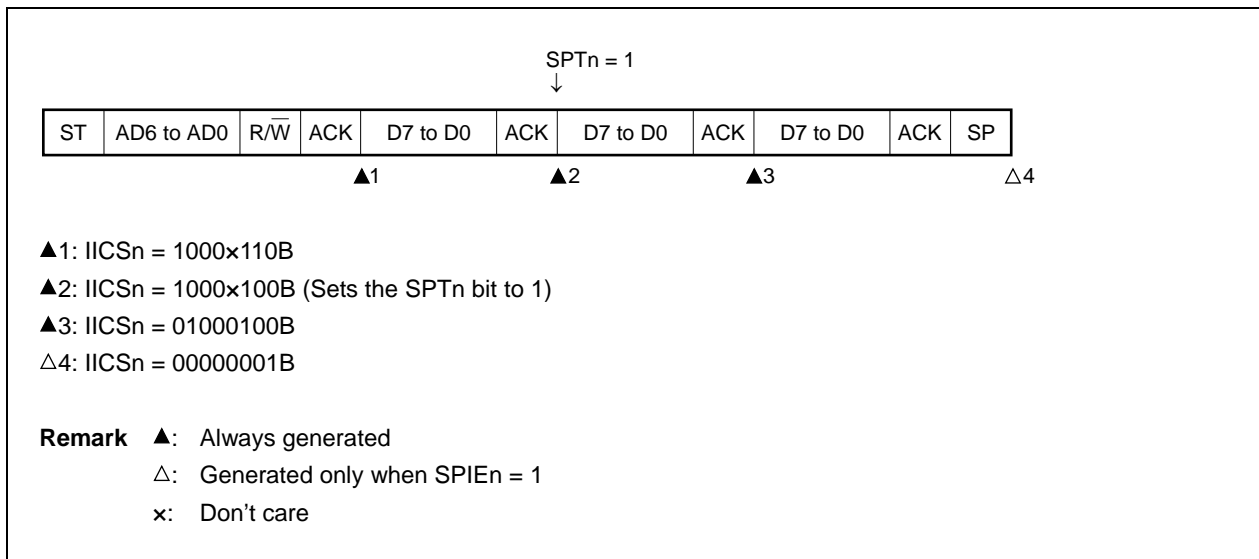
**Remark** n = 0

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

(i) When  $WTIMn = 0$



(ii) When  $WTIMn = 1$



**Remark** n = 0

## 15.6 Timing Charts

When using the I<sup>2</sup>C 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 TRCn bit (bit 3 of the IICA status register n (IICSn)), which specifies the data transfer direction, and then starts serial communication with the slave device.

Figures 15-32 and 15-33 show timing charts of the data communication.

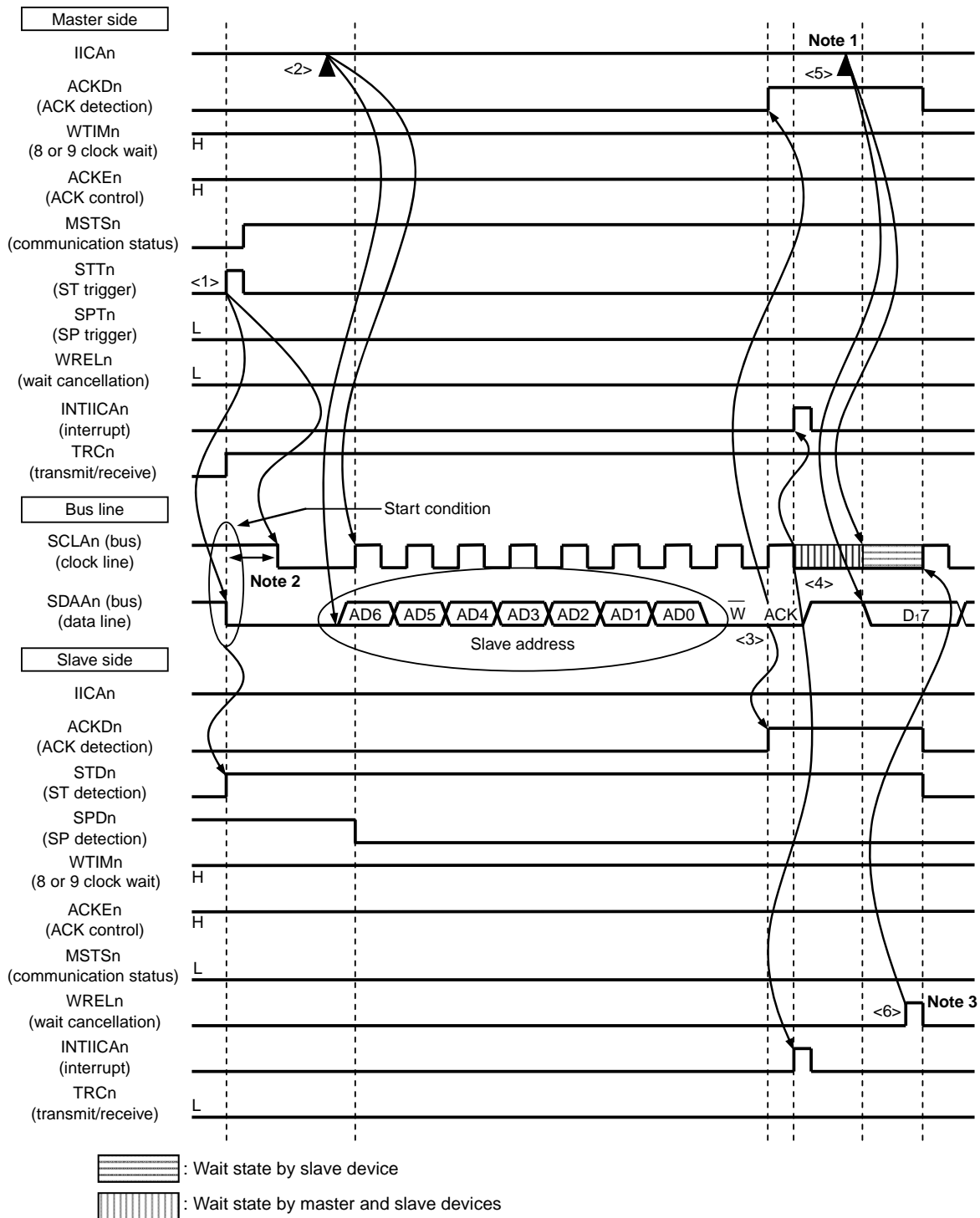
The IICA shift register n (IICAn)'s shift operation is synchronized with the falling edge of the serial clock (SCLAn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAAn pin.

Data input via the SDAAn pin is captured into IICAn at the rising edge of SCLAn.

**Remark** n = 0

**Figure 15-32. Example of Master to Slave Communication  
(9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/4)**

**(1) Start condition ~ address ~ data**



- Notes**
1. Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a master device.
  2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0  $\mu$ s when specifying standard mode and at least 0.6  $\mu$ s when specifying fast mode.
  3. For releasing wait state during reception of a slave device, write "FFH" to IICAn or set the WRELn bit.

**Remark** n = 0

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 15-32 are explained below.

- <1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn = 1 changes SDAAn from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS<sub>n</sub> = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.
- <2> The master device writes the address + W (transmission) to the IICA shift register n (IICAn) and transmits the slave address.
- <3> In the slave device if the address received matches the address (SVAn value) of a slave device<sup>Note</sup>, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLAn = 0) and issues an interrupt (INTIICAn: address match)<sup>Note</sup>.
- <5> The master device writes the data to transmit to the IICAn register and releases the wait status that it set by the master device.
- <6> If the slave device releases the wait status (WRELn = 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: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

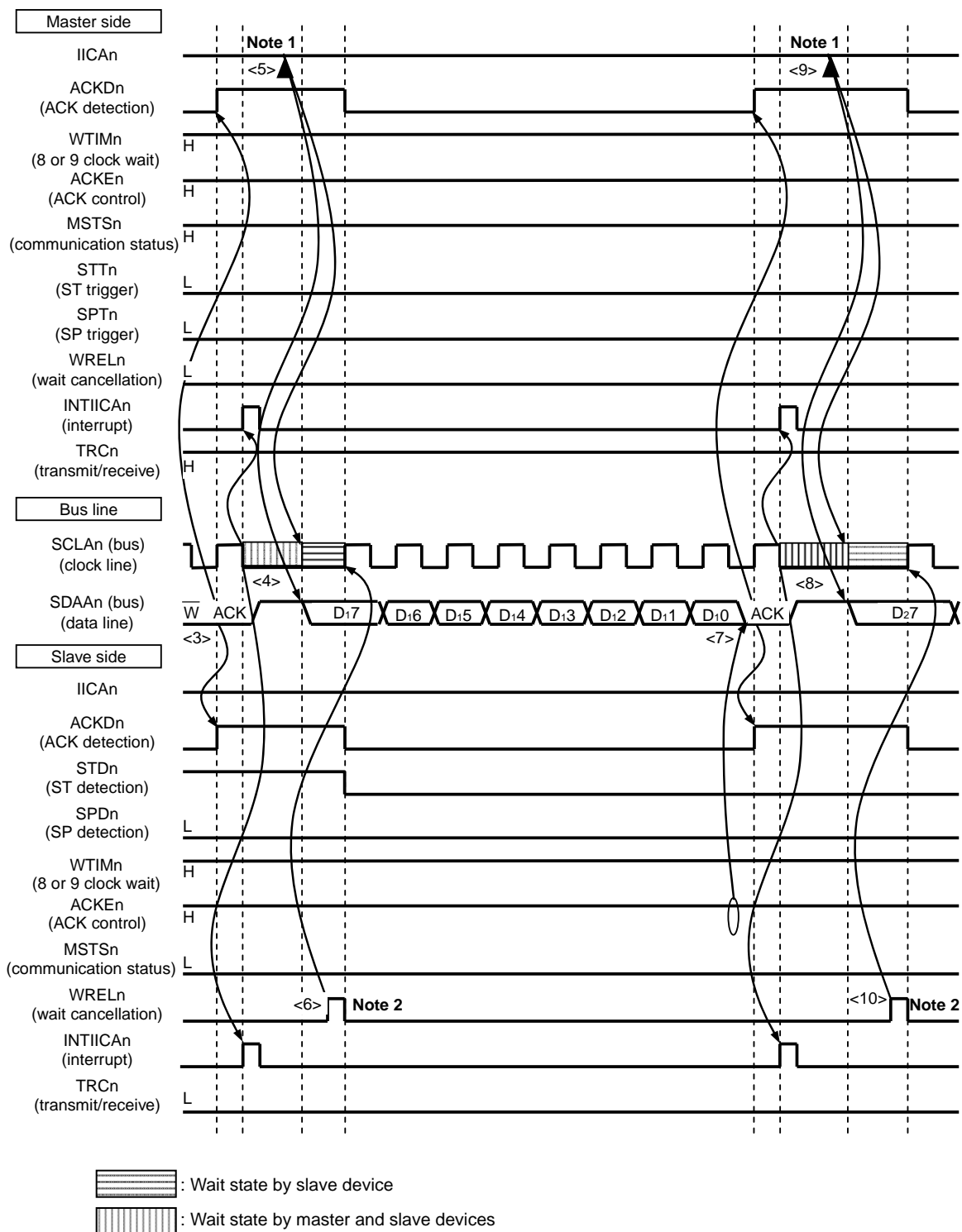
**Remarks 1.** <1> to <15> in Figure 15-32 represent the entire procedure for communicating data using the I<sup>2</sup>C bus.

Figure 15-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

- 2. n = 0

**Figure 15-32. Example of Master to Slave Communication  
(9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/4)**

**(2) Address ~ data ~ data**



- Notes**
1. Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a master device.
  2. For releasing wait state during reception of a slave device, write "FFH" to IICAn or set the WRELn bit.

**Remark** n = 0

The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 15-32 are explained below.

- <3> In the slave device if the address received matches the address (SVAn value) of a slave device<sup>Note</sup>, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLAn = 0) and issues an interrupt (INTIICAn: address match)<sup>Note</sup>.
- <5> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the wait status that it set by the master device.
- <6> If the slave device releases the wait status (WRELn = 1), the master device starts transferring data to the slave device.
- <7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <8> The master device and slave device set a wait status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).
- <9> The master device writes the data to transmit to the IICAn register and releases the wait status that it set by the master device.
- <10> The slave device reads the received data and releases the wait status (WRELn = 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: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

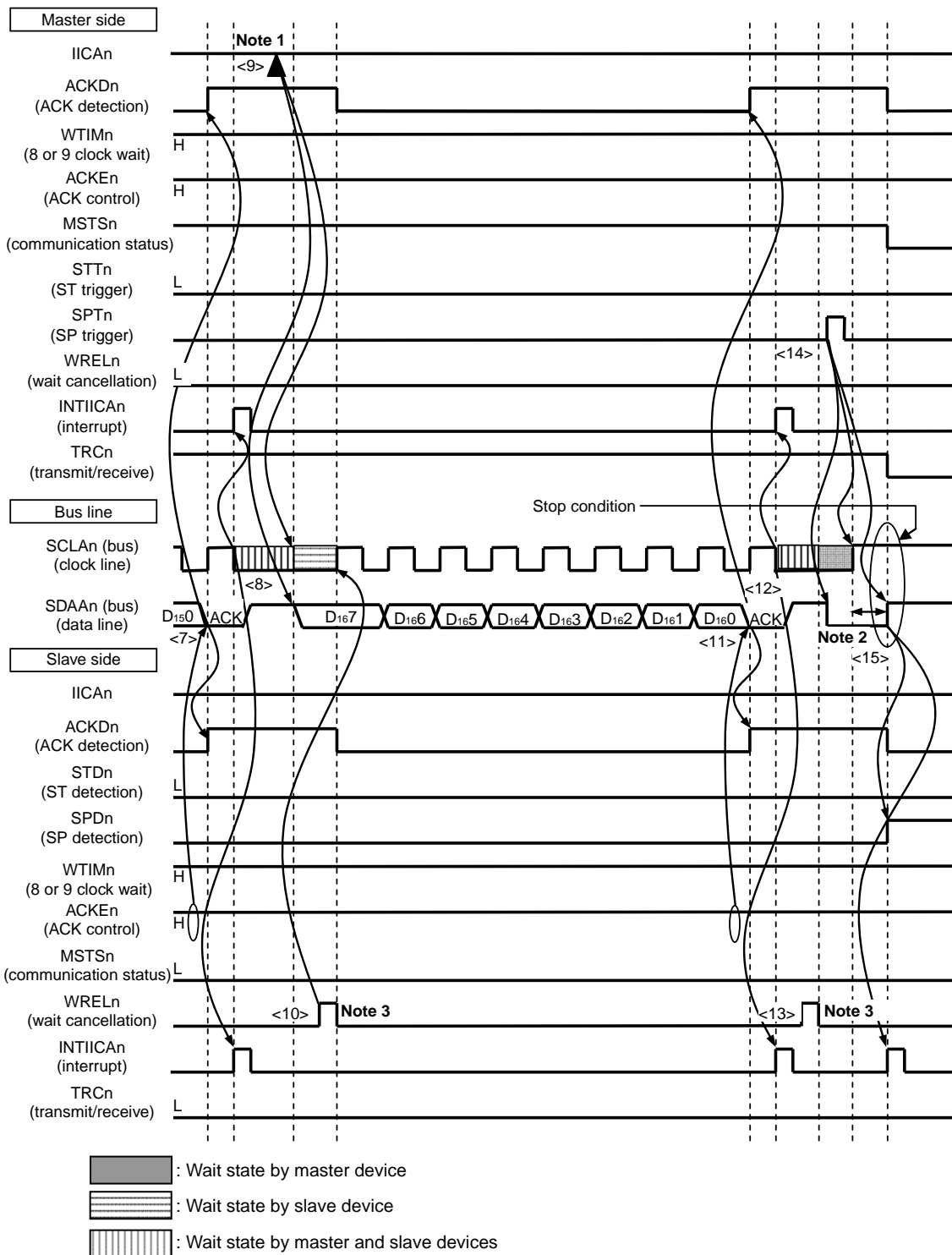
**Remarks 1.** <1> to <15> in Figure 15-32 represent the entire procedure for communicating data using the I<sup>2</sup>C bus.

Figure 15-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

**2.** n = 0

**Figure 15-32. Example of Master to Slave Communication  
(9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/4)**

**(3) Data ~ data ~ Stop condition**



- Notes**
1. Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a master device.
  2. Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0  $\mu$ s when specifying standard mode and at least 0.6  $\mu$ s when specifying fast mode.
  3. For releasing wait state during reception of a slave device, write "FFH" to IICAn or set the WRELn bit.

**Remark** n = 0



The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 15-32 are explained below.

- <7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <8> The master device and slave device set a wait status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).
- <9> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the wait status that it set by the master device.
- <10> The slave device reads the received data and releases the wait status (WRELn = 1). The master device then starts transferring data to the slave device.
- <11> When data transfer is complete, the slave device (ACKEn =1) sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <12> The master device and slave device set a wait status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).
- <13> The slave device reads the received data and releases the wait status (WRELn = 1).
- <14> By the master device setting a stop condition trigger (SPTn = 1), the bus data line is cleared (SDAAn = 0) and the bus clock line is set (SCLAn = 1). After the stop condition setup time has elapsed, by setting the bus data line (SDAAn = 1), the stop condition is then generated (i.e. SCLAn =1 changes SDAAn from 0 to 1).
- <15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICAn: stop condition).

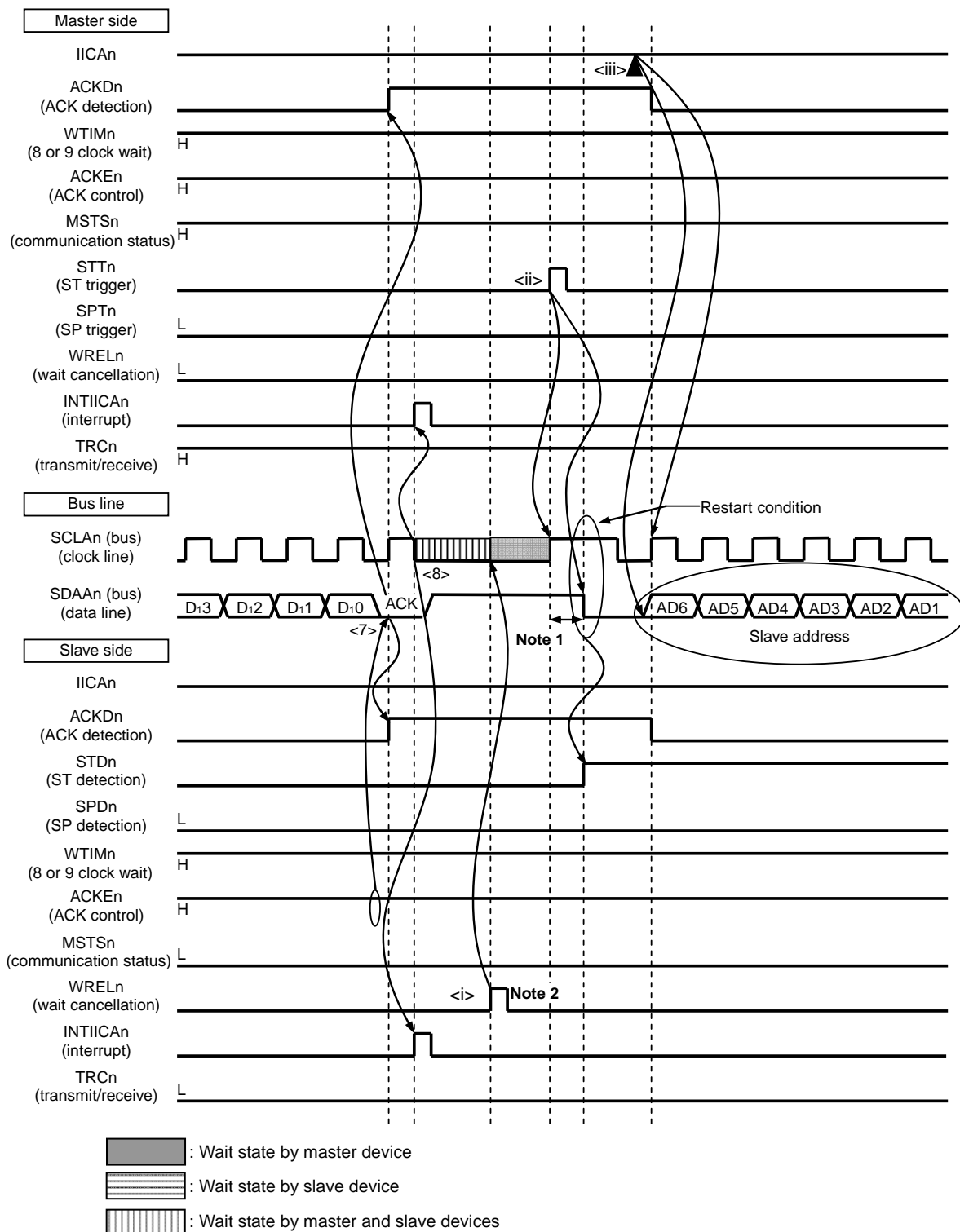
**Remarks 1.** <1> to <15> in Figure 15-32 represent the entire procedure for communicating data using the I<sup>2</sup>C bus.

Figure 15-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

2. n = 0

**Figure 15-32. Example of Master to Slave Communication  
(9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (4/4)**

**(4) Data ~ restart condition ~ address**



**Notes 1.** Make sure that the time between the rise of the SCLAn pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7  $\mu$ s when specifying standard mode and at least 0.6  $\mu$ s when specifying fast mode.

**2.** For releasing wait state during reception of a slave device, write "FFH" to IICAn or set the WRELn bit.

**Remark** n = 0

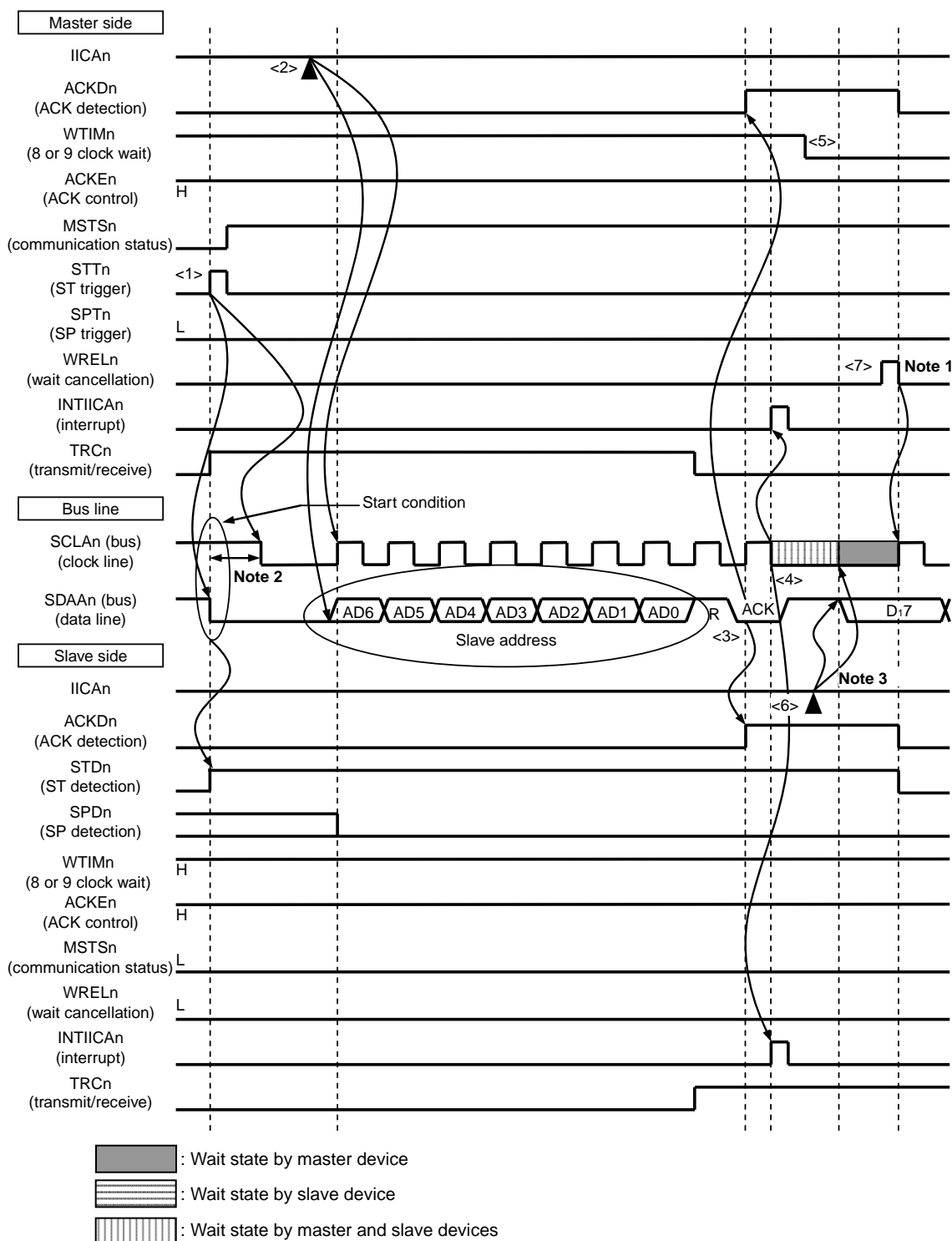
The following describes the operations in Figure 15-32 (4) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <i> to <iii> are performed. These steps return the processing to step <iii>, the data transmission step.

- <7> After data transfer is completed, because of  $ACKEn = 1$ , the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device ( $ACKDn = 1$ ) at the rising edge of the 9th clock.
- <8> The master device and slave device set a wait status ( $SCLAn = 0$ ) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt ( $INTIICAn$ : end of transfer).
- <i> The slave device reads the received data and releases the wait status ( $WRELn = 1$ ).
- <ii> The start condition trigger is set again by the master device ( $STTn = 1$ ) and a start condition (i.e.  $SCLAn = 1$  changes  $SDAAn$  from 1 to 0) is generated once the bus clock line goes high ( $SCLAn = 1$ ) and the bus data line goes low ( $SDAAn = 0$ ) 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 ( $SCLAn = 0$ ) after the hold time has elapsed.
- <iii> The master device writing the address + R/W (transmission) to the IICA shift register ( $IICAn$ ) enables the slave address to be transmitted.

**Remark** n = 0

**Figure 15-33. Example of Slave to Master Communication (8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)**

**(1) Start condition ~ address ~ data**



- Notes**
- For releasing wait state during reception of a master device, write "FFH" to IICAn or set the WRELn bit.
  - Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0  $\mu$ s when specifying standard mode and at least 0.6  $\mu$ s when specifying fast mode.
  - Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a slave device.

**Remark** n = 0

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 15-33 are explained below.

- <1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn = 1 changes SDAAn from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communication status (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.
- <2> The master device writes the address + R (reception) to the IICA shift register n (IICAn) and transmits the slave address.
- <3> In the slave device if the address received matches the address (SVAn value) of a slave device<sup>Note</sup>, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLAn = 0) and issues an interrupt (INTIICAn: address match)<sup>Note</sup>.
- <5> The timing at which the master device sets the wait status changes to the 8th clock (WTIMn = 0).
- <6> The slave device writes the data to transmit to the IICAn register and releases the wait status that it set by the slave device.
- <7> The master device releases the wait status (WRELn = 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: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

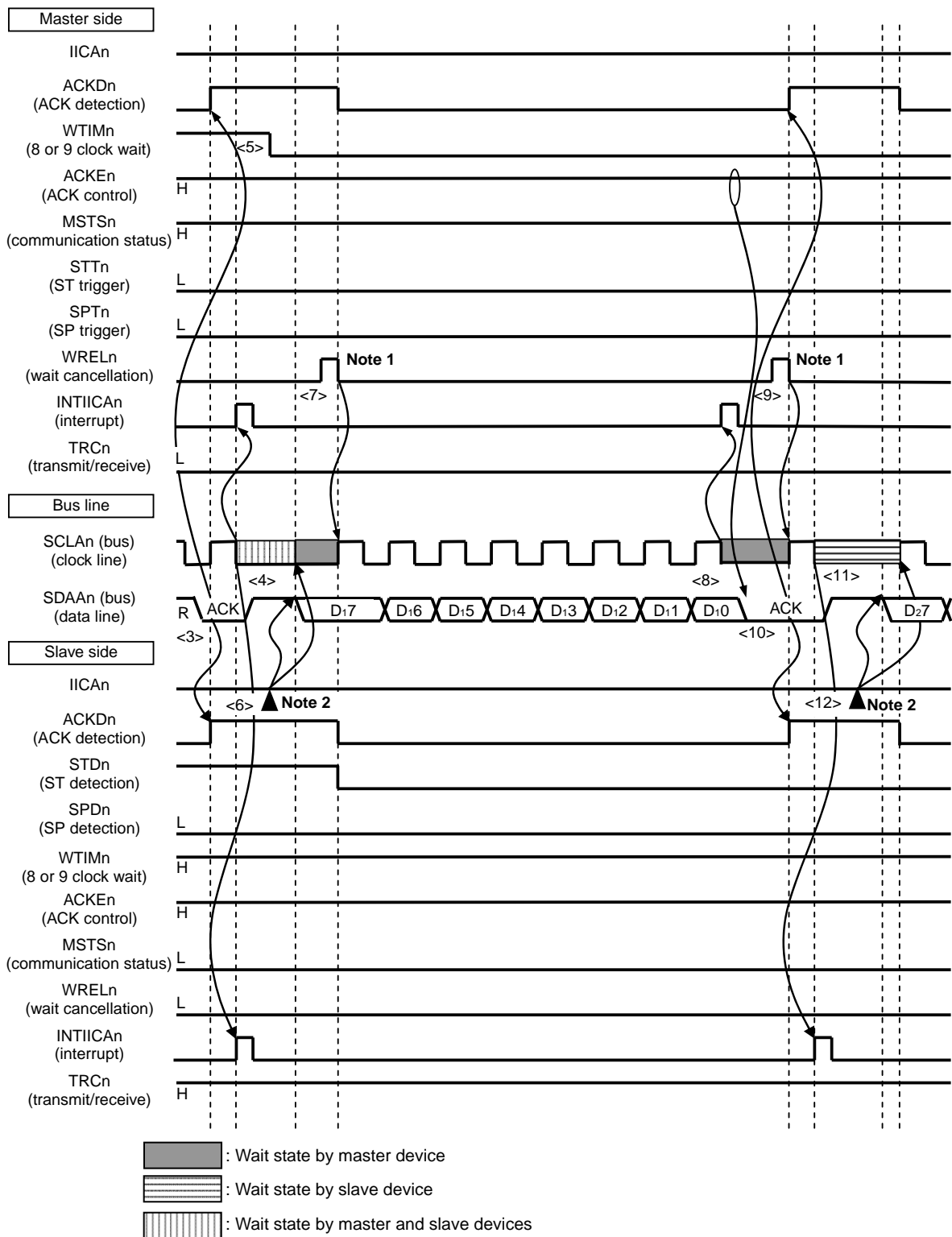
**Remarks 1.** <1> to <19> in Figure 15-33 represent the entire procedure for communicating data using the I<sup>2</sup>C bus.

Figure 15-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 15-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 15-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

- 2. n = 0

**Figure 15-33. Example of Slave to Master Communication  
(8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)**

**(2) Address ~ data ~ data**



- Notes**
- For releasing wait state during reception of a master device, write "FFH" to IICAn or set the WRELn bit.
  - Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a slave device.

**Remark** n = 0

The meanings of <3> to <12> in (2) Address ~ data ~ data in Figure 15-33 are explained below.

- <3> In the slave device if the address received matches the address (SVAn value) of a slave device<sup>Note</sup>, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLAn = 0) and issues an interrupt (INTIICAn: address match)<sup>Note</sup>.
- <5> The master device changes the timing of the wait status to the 8th clock (WTIMn = 0).
- <6> The slave device writes the data to transmit to the IICA shift register n (IICAn) and releases the wait status that it set by the slave device.
- <7> The master device releases the wait status (WRELn = 1) and starts transferring data from the slave device to the master device.
- <8> The master device sets a wait status (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 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 wait status (WRELn = 1).
- <10> The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.
- <11> The slave device set a wait status (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).
- <12> By the slave device writing the data to transmit to the IICAn register, the wait 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: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

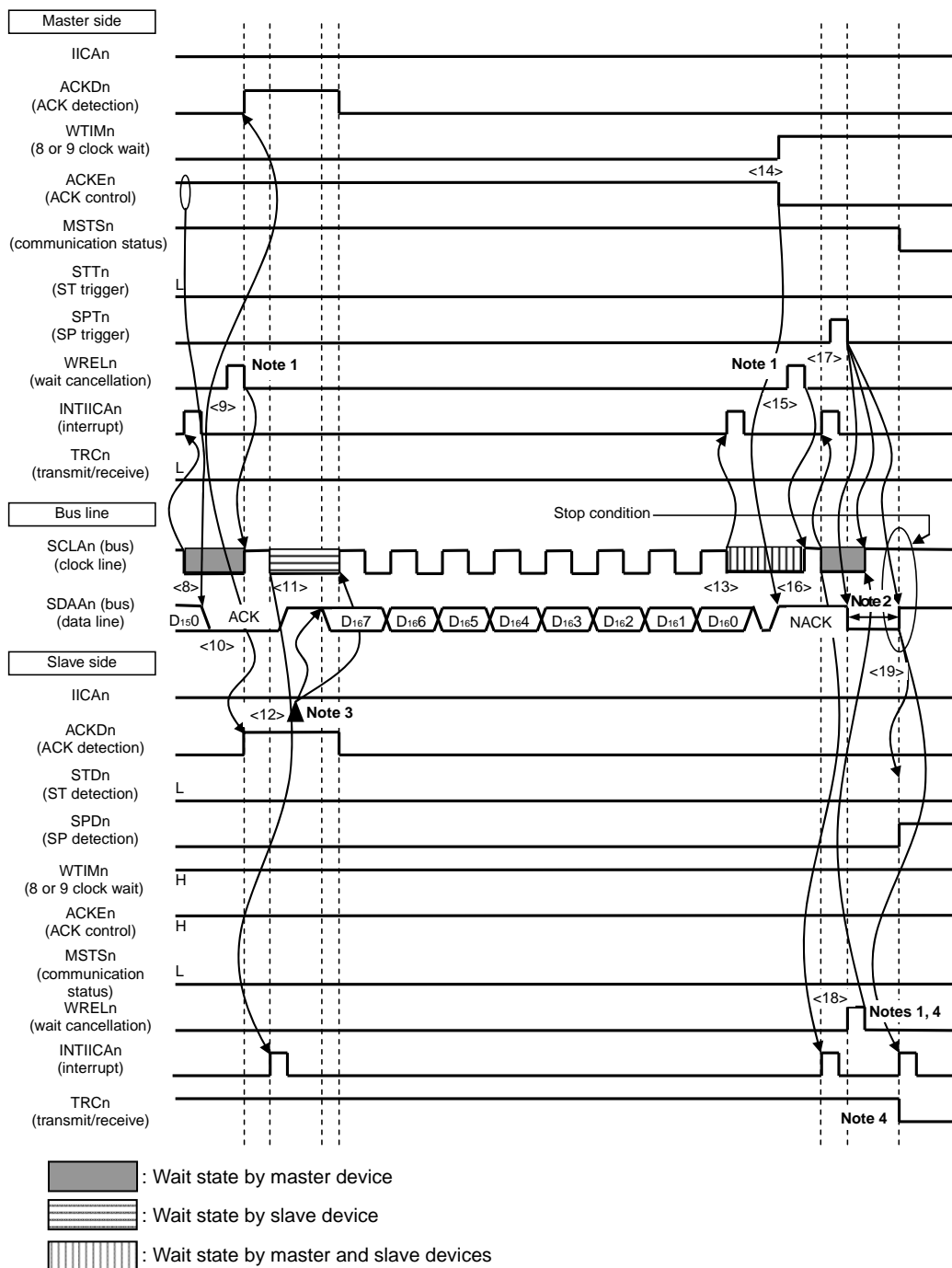
**Remarks 1.** <1> to <19> in Figure 15-33 represent the entire procedure for communicating data using the I<sup>2</sup>C bus.

Figure 15-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 15-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 15-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

2. n = 0

**Figure 15-33. Example of Slave to Master Communication**  
**(8-Clock and 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/3)**

**(3) Data ~ data ~ stop condition**



- Notes**
- To cancel a wait state, write "FFH" to IICAn or set the WRELn bit.
  - Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0  $\mu$ s when specifying standard mode and at least 0.6  $\mu$ s when specifying fast mode.
  - Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a slave device.
  - If a wait state during transmission by a slave device is canceled by setting the WRELn bit, the TRCn bit will be cleared.

**Remark** n = 0



The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 15-33 are explained below.

- <8> The master device sets a wait status (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 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 wait status (WRELn = 1).
- <10> The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.
- <11> The slave device set a wait status (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).
- <12> By the slave device writing the data to transmit to the IICA register, the wait 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 (INTIICAn: end of transfer) at the falling edge of the 8th clock, and sets a wait status (SCLAn = 0). Because ACK control (ACKEn = 1) is performed, the bus data line is at the low level (SDAAn = 0) at this stage.
- <14> The master device sets NACK as the response (ACKEn = 0) and changes the timing at which it sets the wait status to the 9th clock (WTIMn = 1).
- <15> If the master device releases the wait status (WRELn = 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 wait status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).
- <17> When the master device issues a stop condition (SPTn = 1), the bus data line is cleared (SDAAn = 0) and the master device releases the wait status. The master device then waits until the bus clock line is set (SCLAn = 1).
- <18> The slave device acknowledges the NACK, halts transmission, and releases the wait status (WRELn = 1) to end communication. Once the slave device releases the wait status, the bus clock line is set (SCLAn = 1).
- <19> Once the master device recognizes that the bus clock line is set (SCLAn = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDAAn = 1) and issues a stop condition (i.e. SCLAn =1 changes SDAAn from 0 to 1). The slave device detects the generated stop condition and slave device issue an interrupt (INTIICAn: stop condition).

**Remarks 1.** <1> to <19> in Figure 15-33 represent the entire procedure for communicating data using the I<sup>2</sup>C bus.

Figure 15-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 15-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 15-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

**2.** n = 0

CHAPTER 16 LCD CONTROLLER/DRIVER

The number of LCD display function pins of the RL78/L13 differs depending on the product. The following table shows the number of pins of each product.

Table 16-1. Number of LCD Display Function Pins of Each Product

Item	RL78/L13															
	64 pins (R5F10WLx (x = G, F, E, D, C, A))								80 pins (R5F10WMx (x = G, F, E, D, C, A))							
LCD controller/driver	Segment signal outputs: 36 (32) <sup>Note</sup> Common signal outputs: 8								Segment signal outputs: 51 (47) <sup>Note</sup> Common signal outputs: 8							
Multiplexed I/O port	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
P0	SEG 50	SEG 49	SEG 48	SEG 47	SEG 46	SEG 45	SEG 44	SEG 43	SEG 50	SEG 49	SEG 48	SEG 47	SEG 46	SEG 45	SEG 44	SEG 43
P1	SEG 42	SEG 41	SEG 40	SEG 39	SEG 38	SEG 37	SEG 36	SEG 35	SEG 42	SEG 41	SEG 40	SEG 39	SEG 38	SEG 37	SEG 36	SEG 35
P2	SEG 34	SEG 33	-	-	-	SEG 29	-	-	SEG 34	SEG 33	SEG 32	SEG 31	SEG 30	SEG 29	-	-
P3	-	-	-	-	SEG 23	SEG 22	SEG 21	SEG 20	-	-	SEG 25	SEG 24	SEG 23	SEG 22	SEG 21	SEG 20
P4	-	-	-	-	-	-	-	-	SEG 27	SEG 26	-	-	-	-	-	-
P5	SEG 11	-	-	SEG 8	SEG 7	SEG 6	-	-	SEG 11	SEG 10	SEG 9	SEG 8	SEG 7	SEG 6	SEG 5	SEG 4
P7	SEG 19	SEG 18	SEG 17	SEG 16	-	-	-	SEG 12	SEG 19	SEG 18	SEG 17	SEG 16	SEG 15	SEG 14	SEG 13	SEG 12
P13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SEG 28
Not multiplexed with I/O port																
COM4	SEG0								SEG0							
COM5	SEG1								SEG1							
COM6	SEG2								SEG2							
COM7	SEG3								SEG3							

**Note** ( ) indicates the number of signal output pins when 8 com is used.

## 16.1 Functions of LCD Controller/Driver

The functions of the LCD controller/driver in the RL78/L13 microcontrollers are as follows.

- (1) Waveform A or B selectable
- (2) The LCD driver voltage generator can switch internal voltage boosting method, capacitor split method, and external resistance division method.
- (3) Automatic output of segment and common signals based on automatic display data register read
- (4) The reference voltage to be generated when operating the voltage boost circuit can be selected from 16 steps (contrast adjustment).
- (5) LCD blinking is available

Table 16-2 lists the maximum number of pixels that can be displayed in each display mode.

Table 16-2. Maximum Number of Pixels

## (a) 64-pin products

Drive Waveform for LCD Driver	LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform A	External resistance division	–	Static	36 (36 segment signals, 1 common signal)
		1/2	2	72 (36 segment signals, 2 common signals)
			3	108 (36 segment signals, 3 common signals)
		1/3	3	144 (36 segment signals, 4 common signals)
			4	
	1/4	8	256 (32 segment signals, 8 common signals)	
	Internal voltage boosting	1/3	3	108 (36 segment signals, 3 common signals)
			4	144 (36 segment signals, 4 common signals)
		1/4	8	256 (32 segment signals, 8 common signals)
	Capacitor split	1/3	3	108 (36 segment signals, 3 common signals)
			4	144 (36 segment signals, 4 common signals)
	Waveform B	External resistance division, internal voltage boosting	1/3	4
1/4			8	
Capacitor split		1/3	4	144 (36 segment signals, 4 common signals)

## (b) 80-pin products

Drive Waveform for LCD Driver	LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform A	External resistance division	–	Static	51 (51 segment signals, 1 common signal)
		1/2	2	102 (51 segment signals, 2 common signals)
			3	153 (51 segment signals, 3 common signals)
		1/3	3	204 (51 segment signals, 4 common signals)
			4	
	1/4	8	376 (47 segment signals, 8 common signals)	
	Internal voltage boosting	1/3	3	153 (51 segment signals, 3 common signals)
			4	204 (51 segment signals, 4 common signals)
		1/4	8	376 (47 segment signals, 8 common signals)
	Capacitor split	1/3	3	153 (51 segment signals, 3 common signals)
			4	204 (51 segment signals, 4 common signals)
	Waveform B	External resistance division, internal voltage boosting	1/3	4
1/4			8	
Capacitor split		1/3	4	204 (51 segment signals, 4 common signals)

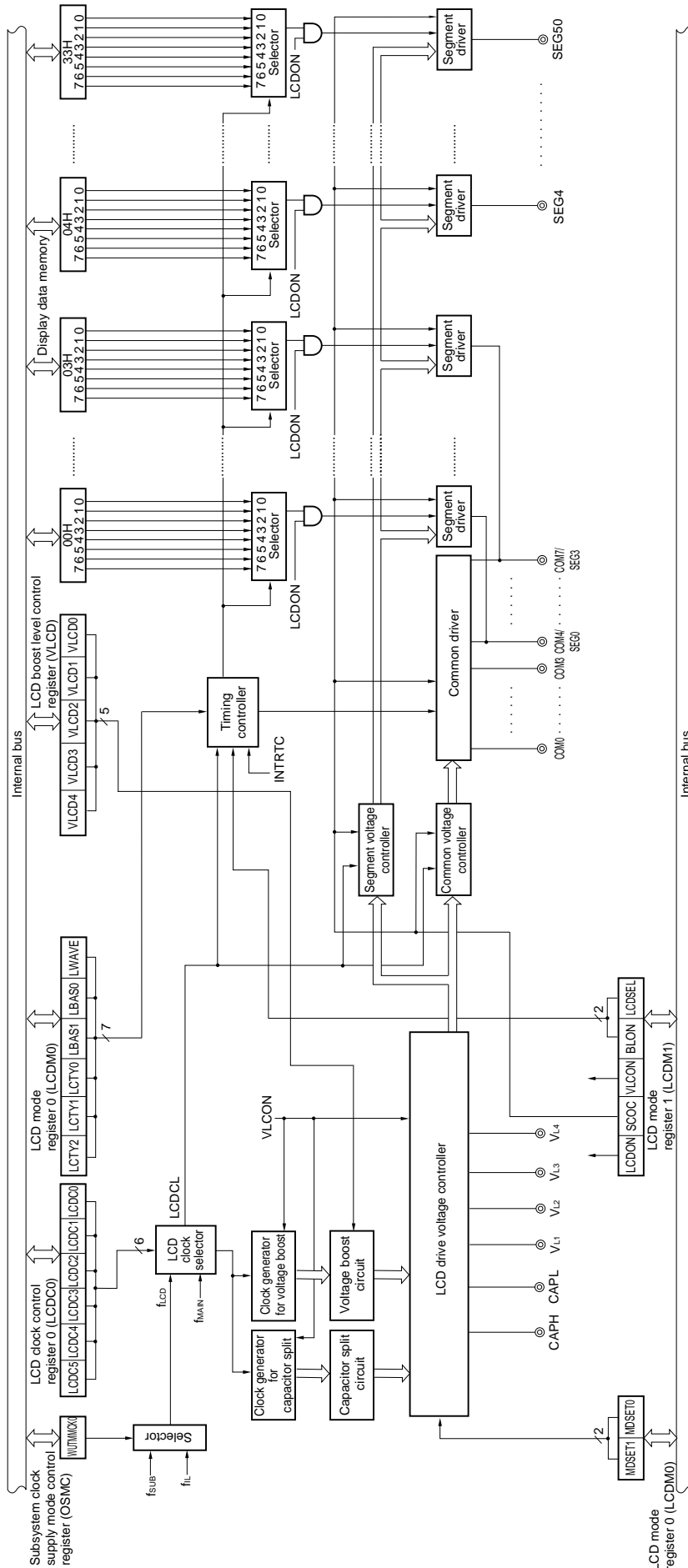
## 16.2 Configuration of LCD Controller/Driver

The LCD controller/driver consists of the following hardware.

**Table 16-3. Configuration of LCD Controller/Driver**

Item	Configuration
Control registers	LCD mode register 0 (LCDM0) LCD mode register 1 (LCDM1) Subsystem clock supply mode control register (OSMC) LCD clock control register 0 (LCDC0) LCD boost level control register (VLCD) LCD input switch control register (ISCLCD) LCD port function registers 0 to 6 (PFSEG0 to PFSEG6) Port mode registers 0 to 5, 7, 13 (PM0 to PM5, PM7, PM13)

Figure 16-1. Block Diagram of LCD Controller/Driver



### 16.3 Registers Controlling LCD Controller/Driver

The following ten registers are used to control the LCD controller/driver.

- LCD mode register 0 (LCDM0)
- LCD mode register 1 (LCDM1)
- Subsystem clock supply mode control register (OSMC)
- LCD clock control register 0 (LCDC0)
- LCD boost level control register (VLCD)
- LCD input switch control register (ISCLCD)
- LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)
- Port mode registers 0 to 5, 7, 13 (PM0 to PM5, PM7, PM13)

**16.3.1 LCD mode register 0 (LCDM0)**

LCDM0 specifies the LCD operation.

This register is set by using an 8-bit memory manipulation instruction.

Reset signal generation sets LCDM0 to 00H.

**Figure 16-2. Format of LCD Mode Register 0 (LCDM0)**

Address: FFF40H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDM0	MDSET1	MDSET0	LWAVE	LDTY2	LDTY1	LDTY0	LBAS1	LBAS0

MDSET1	MDSET0	LCD drive voltage generator selection
0	0	External resistance division method
0	1	Internal voltage boosting method
1	0	Capacitor split method
1	1	Setting prohibited

LWAVE	LCD display waveform selection
0	Waveform A
1	Waveform B

LDTY2	LDTY1	LDTY0	Selection of time slice of LCD display
0	0	0	Static
0	0	1	2-time slice
0	1	0	3-time slice
0	1	1	4-time slice
1	0	1	8-time slice
Other than above			Setting prohibited

LBAS1	LBAS0	LCD display bias mode selection
0	0	1/2 bias method
0	1	1/3 bias method
1	0	1/4 bias method
1	1	Setting prohibited

- Cautions**
1. Do not rewrite the LCDM0 value while the **SCOC** bit of the **LCDM1** register = 1.
  2. When “Static” is selected (LDTY2 to LDTY0 bits = 000B), be sure to set the **LBAS1** and **LBAS0** bits to the default value (00B). Otherwise, the operation will not be guaranteed.
  3. Only the combinations of display waveform, number of time slices, and bias method shown in **Table 16-4** are supported.  
Combinations of settings not shown in **Table 16-4** are prohibited.



Table 16-4. Combinations of Display Waveform, Time Slices, Bias Method, and Frame Frequency

Display Mode			Set Value						Driving Voltage Generation Method		
Display Waveform	Number of Time Slices	Bias Mode	LWAVE	LDTY2	LDTY1	LDTY0	LBAS1	LBAS0	External Resistance Division	Internal Voltage Boosting	Capacitor Split
Waveform A	8	1/4	0	1	0	1	1	0	○ (24 to 128 Hz)	○ (24 to 64 Hz)	×
Waveform A	4	1/3	0	0	1	1	0	1	○ (24 to 128 Hz)	○ (24 to 128 Hz)	○ (24 to 128 Hz)
Waveform A	3	1/3	0	0	1	0	0	1	○ (32 to 128 Hz)	○ (32 to 128 Hz)	○ (32 to 128 Hz)
Waveform A	3	1/2	0	0	1	0	0	0	○ (32 to 128 Hz)	×	×
Waveform A	2	1/2	0	0	0	1	0	0	○ (24 to 128 Hz)	×	×
Waveform A	Static		0	0	0	0	0	0	○ (24 to 128 Hz)	×	×
Waveform B	8	1/4	1	1	0	1	1	0	○ (24 to 128 Hz)	○ (24 to 64 Hz)	×
Waveform B	4	1/3	1	0	1	1	0	1	○ (24 to 128 Hz)	○ (24 to 128 Hz)	○ (24 to 128 Hz)

**Remark** ○: Supported  
 ×: Not supported

### 16.3.2 LCD mode register 1 (LCDM1)

LCDM1 enables or disables display operation, voltage boost circuit operation, and capacitor split circuit operation, and specifies the display data area and the low voltage mode.

LCDM1 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDM1 to 00H.

**Figure 16-3. Format of LCD Mode Register 1 (LCDM1) (1/2)**

Address: FFF41H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
LCDM1	LCDON	SCOC	VLCON	BLON	LCDSEL	0	0	LCDVLM

SCOC	LCDON	LCD display enable/disable
Waveform A or B is output		
0	0	Output ground level to segment/common pin
0	1	
1	0	Display off (all segment outputs are deselected.)
1	1	Display on

VLCON	Voltage boost circuit or capacitor split circuit operation enable/disable
0	Stops voltage boost circuit or capacitor split circuit operation
<sup>1</sup> Note 1	Enables voltage boost circuit or capacitor split circuit operation

BLON <sup>Note 2</sup>	LCDSEL	Display data area control
0	0	Displaying an A-pattern area data (lower four bits of LCD display data register)
0	1	Displaying a B-pattern area data (higher four bits of LCD display data register)
1	0	Alternately displaying A-pattern and B-pattern area data (blinking display corresponding to the constant-period interrupt (INTRTC) timing of real-time clock 2 (RTC2))
1	1	

**Notes** 1. Cannot be set during external resistance division mode.

2. When  $f_{LCD}$  is selected as the LCD source clock ( $f_{LCD}$ ), be sure to set the BLON bit to "0".

(Cautions are listed on the next page.)

Figure 16-3. Format of LCD Mode Register 1 (LCDM1) (2/2)

Address: FFF41H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
LCDM1	LCDON	SCOC	VLCON	BLON	LCDSEL	0	0	LCDVLM

LCDVLM <sup>Note</sup>	Control of default value of voltage boosting pin
0	Set when $V_{DD} \geq 2.7$ V
1	Set when $V_{DD} \leq 4.2$ V

**Note** This function is used to shorten the boost stabilization time by setting the VLx pin to the default status when the voltage boost circuit is used.

If the  $V_{DD}$  voltage is 2.7 V or higher when boosting is started, set the LCDVLM bit to “0”; if the  $V_{DD}$  voltage is 4.2 V or less, set the LCDVLM bit to “1”. However, when  $2.7 \text{ V} \leq V_{DD} \leq 4.2 \text{ V}$ , operation is possible with LCDVLM = 0 or LCDVLM = 1.

- Cautions**
1. When the voltage boost circuit is used, set the SCOC bit = 0 and the VLCON bit = 0, and the MDSET1 and MDSET0 bits = 00B in order to reduce power consumption when the LCD is not used. When the MDSET1 and MDSET0 bits = 01B, power is consumed by the internal reference voltage generator.
  2. When the external resistance division method has been set (MDSET1 and MDSET0 of LCDM0 = 00B) or capacitor split method has been set (MDSET1 and MDSET0 = 10B), set the LCDVLM bit to 0.
  3. Do not rewrite the VLCON and LCDVLM bits while the SCOC bit = 1.
  4. Set the BLON and LCDSEL bits to 0 when 8 has been selected as the number of time slices for the display mode.
  5. To use the internal voltage boosting method, specify the reference voltage by using the VLCD register (select the internal boosting method (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default reference voltage is used), wait for the reference voltage setup time (5 ms (min.)), and then set the VLCON bit to 1.

### 16.3.3 Subsystem clock supply mode control register (OSMC)

OSMC is used to reduce power consumption by stopping as many unnecessary clock functions as possible.

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions, except real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver, is stopped in STOP mode or HALT mode while the subsystem clock is selected as the CPU clock.

In addition, the OSMC register can be used to select the operation clock of real-time clock 2, 12-bit interval timer, clock output/buzzer output, and LCD controller/driver.

This register is set by using an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

**Figure 16-4. 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 subsystem clock supply to peripheral functions. (See <b>Tables 21-1 to 21-3</b> for the peripheral functions whose operations are enabled.)
1	Stops subsystem clock supply to peripheral functions except real-time clock 2, 12-bit interval timer, clock output/buzzer output, and LCD controller/driver.

WUTMMCK0	Selection of operation clock for real-time clock 2, 12-bit interval timer, and LCD controller/driver	Selection of clock output from PCLBUZn pin of clock output/buzzer output controller
0	Subsystem clock ( $f_{SUB}$ )	Selecting the subsystem clock ( $f_{SUB}$ ) is enabled.
1	Low-speed on-chip oscillator clock ( $f_{IL}$ )	Selecting the subsystem clock ( $f_{SUB}$ ) is disabled.

- Cautions**
1. **Be sure to select the subsystem clock (WUTMMCK0 bit = 0) if the subsystem clock is oscillating.**
  2. When WUTMMCK0 is set to 1, the low-speed on-chip oscillator clock oscillates.
  3. **The subsystem clock and low-speed on-chip oscillator clock can only be switched by using the WUTMMCK0 bit if real-time clock 2, 12-bit interval timer, and LCD controller/driver are all stopped.**

**Remark**

- RTCE: Bit 7 of real-time clock control register 0 (RTCC0)
- RINTE: Bit 15 of interval timer control register (ITMC)
- SCOC: Bit 6 of LCD mode register 1 (LCDM1)
- VLCON: Bit 5 of LCD mode register 1 (LCDM1)

**16.3.4 LCD clock control register 0 (LCDC0)**

LCDC0 specifies the LCD clock.

The frame frequency is determined according to the LCD clock and the number of time slices.

This register is set by using an 8-bit memory manipulation instruction.

Reset signal generation sets LCDC0 to 00H.

**Figure 16-5. Format of LCD Clock Control Register 0 (LCDC0)**

Address: FFF42H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDC0	0	0	LCDC05	LCDC04	LCDC03	LCDC02	LCDC01	LCDC00

LCDC05	LCDC04	LCDC03	LCDC02	LCDC01	LCDC00	LCD clock (LCDCL)
0	0	0	1	0	0	$f_{SUB}/2^5$ or $f_{IL}/2^5$
0	0	0	1	0	1	$f_{SUB}/2^6$ or $f_{IL}/2^6$
0	0	0	1	1	0	$f_{SUB}/2^7$ or $f_{IL}/2^7$
0	0	0	1	1	1	$f_{SUB}/2^8$ or $f_{IL}/2^8$
0	0	1	0	0	0	$f_{SUB}/2^9$ or $f_{IL}/2^9$
0	0	1	0	0	1	$f_{SUB}/2^{10}$
0	1	0	0	1	1	$f_{MAIN}/2^{10}$
0	1	0	1	0	0	$f_{MAIN}/2^{11}$
0	1	0	1	0	1	$f_{MAIN}/2^{12}$
0	1	0	1	1	0	$f_{MAIN}/2^{13}$
0	1	0	1	1	1	$f_{MAIN}/2^{14}$
0	1	1	0	0	0	$f_{MAIN}/2^{15}$
0	1	1	0	0	1	$f_{MAIN}/2^{16}$
0	1	1	0	1	0	$f_{MAIN}/2^{17}$
0	1	1	0	1	1	$f_{MAIN}/2^{18}$
1	0	1	0	1	1	$f_{MAIN}/2^{19}$
Other than above						Setting prohibited

- Cautions**
1. Do not set LCDC0 when the SCOC bit of the LCDM1 register is 1.
  2. Be sure to set bits 6 and 7 to "0".
  3. When the internal voltage boosting method or capacitor split method is set, set the LCD clock (LCDCL) as follows:
    - 512 Hz or less when  $f_{SUB}$  is selected.
    - 235 Hz or less when  $f_{IL}$  is selected.
- For details, see Table 16-4. Combinations of Display Waveform, Time Slices, Bias Method, and Frame Frequency.

**Remark**  $f_{MAIN}$ : Main system clock frequency  
 $f_{IL}$ : Low-speed on-chip oscillator clock frequency  
 $f_{SUB}$ : Subsystem clock frequency

**16.3.5 LCD boost level control register (VLCD)**

VLCD selects the reference voltage that is to be generated when operating the voltage boost circuit (contrast adjustment). The reference voltage can be selected from 16 steps.

This register is set by using an 8-bit memory manipulation instruction.

Reset signal generation sets VLCD to 04H.

**Figure 16-6. Format of LCD Boost Level Control Register (VLCD)**

Address: FFF43H After reset: 04H R/W

Symbol	7	6	5	4	3	2	1	0
VLCD	0	0	0	VLCD4	VLCD3	VLCD2	VLCD1	VLCD0

VLCD4	VLCD3	VLCD2	VLCD1	VLCD0	Reference voltage selection (contrast adjustment)	VL <sub>L</sub> voltage	
						1/3 bias method	1/4 bias method
0	0	1	0	0	1.00 V (default)	3.00 V	4.00 V
0	0	1	0	1	1.05 V	3.15 V	4.20 V
0	0	1	1	0	1.10 V	3.30 V	4.40 V
0	0	1	1	1	1.15 V	3.45 V	4.60 V
0	1	0	0	0	1.20 V	3.60 V	4.80 V
0	1	0	0	1	1.25 V	3.75 V	5.00 V
0	1	0	1	0	1.30 V	3.90 V	5.20 V
0	1	0	1	1	1.35 V	4.05 V	Setting prohibited
0	1	1	0	0	1.40 V	4.20 V	Setting prohibited
0	1	1	0	1	1.45 V	4.35 V	Setting prohibited
0	1	1	1	0	1.50 V	4.50 V	Setting prohibited
0	1	1	1	1	1.55 V	4.65 V	Setting prohibited
1	0	0	0	0	1.60 V	4.80 V	Setting prohibited
1	0	0	0	1	1.65 V	4.95 V	Setting prohibited
1	0	0	1	0	1.70 V	5.10 V	Setting prohibited
1	0	0	1	1	1.75 V	5.25 V	Setting prohibited
Other than above					Setting prohibited		

- Cautions**
1. The VLCD setting is valid only when the voltage boost circuit is operating.
  2. Be sure to set bits 5 to 7 to “0”.
  3. Be sure to change the VLCD value after having stopped the operation of the voltage boost circuit (VLCON = 0).
  4. To use the internal voltage boosting method, specify the reference voltage by using the VLCD register (select the internal boosting method (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default reference voltage is used), wait for the reference voltage setup time (5 ms (min.)), and then set VLCON to 1.
  5. To use the external resistance division method or capacitor split method, use the VLCD register with its initial value (04H).

**16.3.6 LCD input switch control register (ISCLCD)**

Input to the Schmitt trigger buffer must be disabled until the CAPL/P126, CAPH/P127, and VL3/P125 pins are set to operate as LCD function pins in order to prevent through-current from entering.

This register is set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets ISCLCD to 00H.

**Figure 16-7. Format of LCD Input Switch Control Register (ISCLCD)**

Address: F0308H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISCLCD	0	0	0	0	0	0	ISCVL3	ISCCAP

ISCVL3	VL3/P125 pin Schmitt trigger buffer control
0	Input invalid
1	Input valid

ISCCAP	CAPL/P126, CAPH/P127 pins Schmitt trigger buffer control
0	Input invalid
1	Input valid

**Caution** If ISCVL3 = 0 and ISCCAP = 0, set the corresponding port registers as follows:

**PU127 bit of PU12 register = 0, P127 bit of P12 register = 0**

**PU126 bit of PU12 register = 0, P126 bit of P12 register = 0**

**PU125 bit of PU12 register = 0, P125 bit of P12 register = 0**

**(a) Operation of ports that alternately function as VL3, CAPL, and CAPH pins**

The functions of the VL3/P125, CAPL/P126, and CAPH/P127 pins can be selected by using the LCD input switch control register (ISCLCD), LCD mode register 0 (LCDM0), and port mode register 12 (PM12).

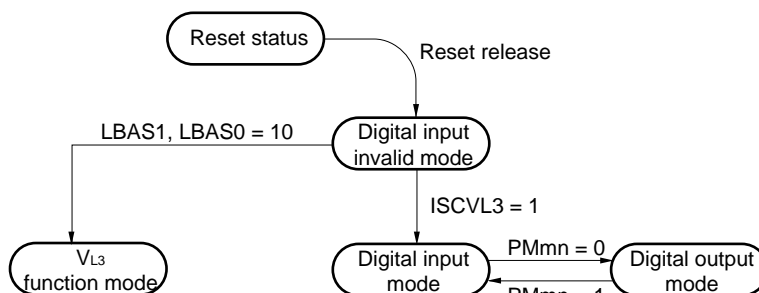
- VL3/P125

**Table 16-5. Settings of VL3/P125 Pin Function**

Bias Setting (LBAS1 and LBAS0 Bits of LCDM0 Register )	ISCVL3 Bit of ISCLCD Register	PM125 Bit of PM12 Register	Pin Function	Initial Status
Other than 1/4 bias method (LBAS1, LBAS0 = 00 or 01)	0	1	Digital input invalid mode	√
	1	0	Digital output mode	–
	1	1	Digital input mode	–
1/4 bias method (LBAS1, LBAS0 = 10)	0	1	VL3 function mode	–
Other than above			Setting prohibited	

The following shows the VL3/P125 pin function status transitions.

**Figure 16-8. VL3/P125 Pin Function Status Transitions**



**Caution** Be sure to set the VL3 function mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

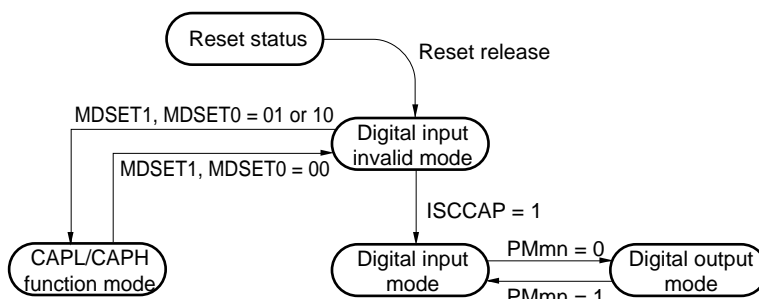
• CAPL/P126 and CAPH/P127

**Table 16-6. Settings of CAPL/P126 and CAPH/P127 Pin Functions**

LCD Drive Voltage Generator (MDSET1 and MDSET0 Bits of LCDM0 Register)	ISCCAP Bit of ISCLCD Register	PM126 and PM127 Bits of PM12 Register	Pin Function	Initial Status
External resistance division (MDSET1, MDSET0 = 00)	0	1	Digital input invalid mode	√
	1	0	Digital output mode	–
	1	1	Digital input mode	–
Internal voltage boosting or capacitor split (MDSET1, MDSET0 = 01 or 10)	0	1	CAPL/CAPH function mode	–
Other than above			Setting prohibited	

The following shows the CAPL/P126 and CAPH/P127 pin function status transitions.

**Figure 16-9. CAPL/P126 and CAPH/P127 Pin Function Status Transitions**



**Caution** Be sure to set the CAPL/CAPH function mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).



### 16.3.7 LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)

These registers set whether to use pins P00 to P07, P10 to P17, P22 to P27, P30 to P35, P46, P47, P50 to P57, P70 to P77, and P130 as port pins (or alternate function pins) or segment output, and whether to use P45 as a port pin or an analog input pin.

These registers are set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH (PFSEG0 is F0H, PFSEG6 is 0FH).

**Remark** The correspondence between the segment output pins (SEGxx) and the PFSEG register (PFSEGxx bits) and the existence of SEGxx pins in each product are shown in **Table 16-7 Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits)**.

**Figure 16-10. Format of LCD Port Function Registers 0 to 6 (80-pin Products)**

Address: F0300H After reset: F0H R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG0	PFSEG07	PFSEG06	PFSEG05	PFSEG04	0	0	0	0

Address: F0301H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG1	PFSEG15	PFSEG14	PFSEG13	PFSEG12	PFSEG11	PFSEG10	PFSEG09	PFSEG08

Address: F0302H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG2	PFSEG23	PFSEG22	PFSEG21	PFSEG20	PFSEG19	PFSEG18	PFSEG17	PFSEG16

Address: F0303H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG3	PFSEG30	PFSEG29	PFSEG28	PFSEG27	PFSEG26	PFDEG	PFSEG25	PFSEG24

Address: F0304H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG4	PFSEG38	PFSEG37	PFSEG36	PFSEG35	PFSEG34	PFSEG33	PFSEG32	PFSEG31

Address: F0305H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG5	PFSEG46	PFSEG45	PFSEG44	PFSEG43	PFSEG42	PFSEG41	PFSEG40	PFSEG39

Address: F0306H After reset: 0FH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG6	0	0	0	0	PFSEG50	PFSEG49	PFSEG48	PFSEG47

PFSEGxx (xx = 04 to 50)	Specification of port (other than segment output)/segment output for Pmn pins (mn = 00 to 07, 10 to 17, 22 to 27, 30 to 35, 46, 47, 50 to 57, 70 to 77, 130)
0	Used as port (other than segment output)
1	Used as segment output

PFDEG	Specification of port/analog input (IVREF0) for P45 pin
0	Used as port
1	Used as analog input (IVREF0)

**Remark** To use the Pmn pins as segment output pins (PFSEGxx = 1), be sure to set the PUmn bit of the PUm register, POMmn bit of the POMm register, and PIMmn bit of the PIMm register to "0".

**Table 16-7. Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits)**

Bit name of PFSEG Register	Corresponding SEGxx Pins	Alternate Port	80-pin	64-pin
PFSEG04	SEG4	P50	√	–
PFSEG05	SEG5	P51	√	–
PFSEG06	SEG6	P52	√	√
PFSEG07	SEG7	P53	√	√
PFSEG08	SEG8	P54	√	√
PFSEG09	SEG9	P55	√	–
PFSEG10	SEG10	P56	√	–
PFSEG11	SEG11	P57	√	√
PFSEG12	SEG12	P70	√	√
PFSEG13	SEG13	P71	√	–
PFSEG14	SEG14	P72	√	–
PFSEG15	SEG15	P73	√	–
PFSEG16	SEG16	P74	√	√
PFSEG17	SEG17	P75	√	√
PFSEG18	SEG18	P76	√	√
PFSEG19	SEG19	P77	√	√
PFSEG20	SEG20	P30	√	√
PFSEG21	SEG21	P31	√	√
PFSEG22	SEG22	P32	√	√
PFSEG23	SEG23	P33	√	√
PFSEG24	SEG24	P34	√	–
PFSEG25	SEG25	P35	√	–
PFSEG26	SEG26	P46	√	–
PFSEG27	SEG27	P47	√	–
PFSEG28	SEG28	P130	√	–
PFSEG29	SEG29	P22	√	√
PFSEG30	SEG30	P23	√	–
PFSEG31	SEG31	P24	√	–
PFSEG32	SEG32	P25	√	–
PFSEG33	SEG33	P26	√	√
PFSEG34	SEG34	P27	√	√
PFSEG35	SEG35	P10	√	√
PFSEG36	SEG36	P11	√	√
PFSEG37	SEG37	P12	√	√
PFSEG38	SEG38	P13	√	√
PFSEG39	SEG39	P14	√	√
PFSEG40	SEG40	P15	√	√
PFSEG41	SEG41	P16	√	√
PFSEG42	SEG42	P17	√	√
PFSEG43	SEG43	P00	√	√
PFSEG44	SEG44	P01	√	√
PFSEG45	SEG45	P02	√	√
PFSEG46	SEG46	P03	√	√
PFSEG47	SEG47	P04	√	√
PFSEG48	SEG48	P05	√	√
PFSEG49	SEG49	P06	√	√
PFSEG50	SEG50	P07	√	√

**(a) Operation of ports that alternately function as SEGxx pins**

The functions of ports that also serve as segment output pins (SEGxx) can be selected by using the port mode control register (PMCxx), port mode register (PMxx), and LCD port function registers 0 to 6 (PFSEG0 to PFSEG6).

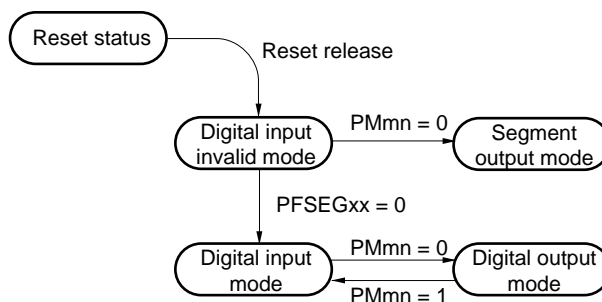
- P00 to P07, P14 to P17, P30 to P35, P46, P47, P50 to P57, P70 to P77, P130 (ports that do not serve as analog input pins (ANLxx))

**Table 16-8. Settings of SEGxx/Port Pin Function**

PFSEGxx Bit of PFSEG0 to PFSEG6 Registers	PMxx Bit of PMxx Register	Pin Function	Initial Status
1	1	Digital input invalid mode	√
0	0	Digital output mode	–
0	1	Digital input mode	–
1	0	Segment output mode	–

The following shows the SEGxx/port pin function status transitions.

**Figure 16-11. SEGxx/Port Pin Function Status Transitions**



**Caution** Be sure to set the segment output mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

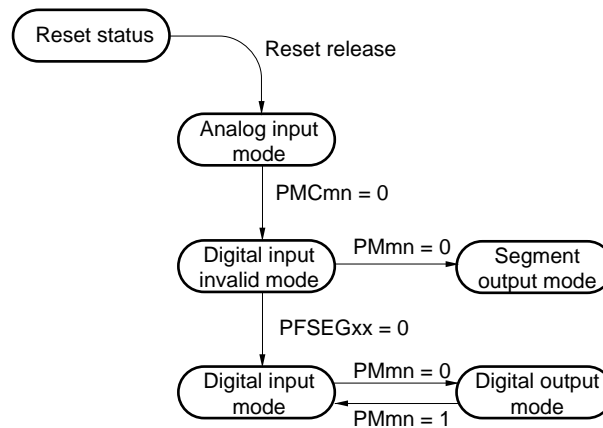
- P10 to P13, P22 to P27 (ports that serve as analog input pins (ANLxx))

**Table 16-9. Settings of ANLxx/SEGxx/Port Pin Function**

PMCxx Bit of PMCxx Register	PFSEGxx Bit of PFSEG3 and PFSEG4 Registers	PMxx Bit of PMxx Register	Pin Function	Initial Status
1	1	1	Analog input mode	√
0	0	0	Digital output mode	–
0	0	1	Digital input mode	–
0	1	0	Segment output mode	–
0	1	1	Digital input invalid mode	–
Other than above			Setting prohibited	

The following shows the ANLxx/SEGxx/port pin function status transitions.

**Figure 16-12. ANLxx/SEGxx/Port Pin Function Status Transitions**



**Caution** Be sure to set the segment output mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

**16.3.8 Port mode registers 0 to 5, 7, 13 (PM0 to PM5, PM7, PM13)**

These registers specify input/output of ports 0 to 5, 7, and 13 in 1-bit units.

When using the ports (such as P00/SEG43/SO00/TxD0/TOOLTxD) to be shared with the segment output pin for segment output, set the port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

**Example:** When using P00/SEG43/SO00/TxD0/TOOLTxD for segment output  
 Set the PM00 bit of port mode register 0 to “0”.  
 Set the P00 bit of port register 0 to “0”.

These registers are set by using a 1-bit or 8-bit memory manipulation instruction.  
 Reset signal generation sets these registers to FFH.

**Figure 16-13. Format of Port Mode Registers 0 to 5, 7, 13 (PM0 to PM5, PM7, PM13) (80-pin Products)**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	PM35	PM34	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM13	1	1	1	1	1	1	1	PM130	FFF2DH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 5, 7, 13; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

**Remark** The figure shown above presents the format of port mode registers 0 to 5, 7, and 13 of the 80-pin products. The format of the port mode register of other products, see **Table 4-3 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product.**

## 16.4 LCD Display Data Registers

The LCD display data registers are mapped as shown in Table 16-10. The contents displayed on the LCD can be changed by changing the contents of the LCD display data registers.

**Table 16-10. Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (1/4)**

**(a) Other than 8-time-slice (static, 2-time slice, 3-time slice, and 4-time slice) (1/2)**

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	80-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0		
SEG0	F0400H	SEG0 (B-pattern area)				SEG0 (A-pattern area)				√	√
SEG1	F0401H	SEG1 (B-pattern area)				SEG1 (A-pattern area)				√	√
SEG2	F0402H	SEG2 (B-pattern area)				SEG2 (A-pattern area)				√	√
SEG3	F0403H	SEG3 (B-pattern area)				SEG3 (A-pattern area)				√	√
SEG4	F0404H	SEG4 (B-pattern area)				SEG4 (A-pattern area)				√	–
SEG5	F0405H	SEG5 (B-pattern area)				SEG5 (A-pattern area)				√	–
SEG6	F0406H	SEG6 (B-pattern area)				SEG6 (A-pattern area)				√	√
SEG7	F0407H	SEG7 (B-pattern area)				SEG7 (A-pattern area)				√	√
SEG8	F0408H	SEG8 (B-pattern area)				SEG8 (A-pattern area)				√	√
SEG9	F0409H	SEG9 (B-pattern area)				SEG9 (A-pattern area)				√	–
SEG10	F040AH	SEG10 (B-pattern area)				SEG10 (A-pattern area)				√	–
SEG11	F040BH	SEG11 (B-pattern area)				SEG11 (A-pattern area)				√	√
SEG12	F040CH	SEG12 (B-pattern area)				SEG12 (A-pattern area)				√	√
SEG13	F040DH	SEG13 (B-pattern area)				SEG13 (A-pattern area)				√	–
SEG14	F040EH	SEG14 (B-pattern area)				SEG14 (A-pattern area)				√	–
SEG15	F040FH	SEG15 (B-pattern area)				SEG15 (A-pattern area)				√	–
SEG16	F0410H	SEG16 (B-pattern area)				SEG16 (A-pattern area)				√	√
SEG17	F0411H	SEG17 (B-pattern area)				SEG17 (A-pattern area)				√	√
SEG18	F0412H	SEG18 (B-pattern area)				SEG18 (A-pattern area)				√	√
SEG19	F0413H	SEG19 (B-pattern area)				SEG19 (A-pattern area)				√	√
SEG20	F0414H	SEG20 (B-pattern area)				SEG20 (A-pattern area)				√	√
SEG21	F0415H	SEG21 (B-pattern area)				SEG21 (A-pattern area)				√	√
SEG22	F0416H	SEG22 (B-pattern area)				SEG22 (A-pattern area)				√	√
SEG23	F0417H	SEG23 (B-pattern area)				SEG23 (A-pattern area)				√	√
SEG24	F0418H	SEG24 (B-pattern area)				SEG24 (A-pattern area)				√	–
SEG25	F0419H	SEG25 (B-pattern area)				SEG25 (A-pattern area)				√	–
Use prohibited	F041AH	Use prohibited									
SEG26	F041BH	SEG26 (B-pattern area)				SEG26 (A-pattern area)				√	–
SEG27	F041CH	SEG27 (B-pattern area)				SEG27 (A-pattern area)				√	–
SEG28	F041DH	SEG28 (B-pattern area)				SEG28 (A-pattern area)				√	–
SEG29	F041EH	SEG29 (B-pattern area)				SEG29 (A-pattern area)				√	√
SEG30	F041FH	SEG30 (B-pattern area)				SEG30 (A-pattern area)				√	–
SEG31	F0420H	SEG31 (B-pattern area)				SEG31 (A-pattern area)				√	–
SEG32	F0421H	SEG32 (B-pattern area)				SEG32 (A-pattern area)				√	–

**Table 16-10. Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (2/4)****(a) Other than 8-time-slice (static, 2-time slice, 3-time slice, and 4-time slice) (2/2)**

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	80-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0		
SEG33	F0422H	SEG33 (B-pattern area)				SEG33 (A-pattern area)				√	√
SEG34	F0423H	SEG34 (B-pattern area)				SEG34 (A-pattern area)				√	√
SEG35	F0424H	SEG35 (B-pattern area)				SEG35 (A-pattern area)				√	√
SEG36	F0425H	SEG36 (B-pattern area)				SEG36 (A-pattern area)				√	√
SEG37	F0426H	SEG37 (B-pattern area)				SEG37 (A-pattern area)				√	√
SEG38	F0427H	SEG38 (B-pattern area)				SEG38 (A-pattern area)				√	√
SEG39	F0428H	SEG39 (B-pattern area)				SEG39 (A-pattern area)				√	√
SEG40	F0429H	SEG40 (B-pattern area)				SEG40 (A-pattern area)				√	√
SEG41	F042AH	SEG41 (B-pattern area)				SEG41 (A-pattern area)				√	√
SEG42	F042BH	SEG42 (B-pattern area)				SEG42 (A-pattern area)				√	√
SEG43	F042CH	SEG43 (B-pattern area)				SEG43 (A-pattern area)				√	√
SEG44	F042DH	SEG44 (B-pattern area)				SEG44 (A-pattern area)				√	√
SEG45	F042EH	SEG45 (B-pattern area)				SEG45 (A-pattern area)				√	√
SEG46	F042FH	SEG46 (B-pattern area)				SEG46 (A-pattern area)				√	√
SEG47	F0430H	SEG47 (B-pattern area)				SEG47 (A-pattern area)				√	√
SEG48	F0431H	SEG48 (B-pattern area)				SEG48 (A-pattern area)				√	√
SEG49	F0432H	SEG49 (B-pattern area)				SEG49 (A-pattern area)				√	√
SEG50	F0433H	SEG50 (B-pattern area)				SEG50 (A-pattern area)				√	√

**Remark** √: Supported, -: Not supported



Table 16-10. Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (3/4)

## (b) 8-time slice (1/2)

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	80-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0		
SEG0	F0400H	SEG0 <sup>Note</sup>								√	√
SEG1	F0401H	SEG1 <sup>Note</sup>								√	√
SEG2	F0402H	SEG2 <sup>Note</sup>								√	√
SEG3	F0403H	SEG3 <sup>Note</sup>								√	√
SEG4	F0404H	SEG4								√	–
SEG5	F0405H	SEG5								√	–
SEG6	F0406H	SEG6								√	√
SEG7	F0407H	SEG7								√	√
SEG8	F0408H	SEG8								√	√
SEG9	F0409H	SEG9								√	–
SEG10	F040AH	SEG10								√	–
SEG11	F040BH	SEG11								√	√
SEG12	F040CH	SEG12								√	√
SEG13	F040DH	SEG13								√	–
SEG14	F040EH	SEG14								√	–
SEG15	F040FH	SEG15								√	–
SEG16	F0410H	SEG16								√	√
SEG17	F0411H	SEG17								√	√
SEG18	F0412H	SEG18								√	√
SEG19	F0413H	SEG19								√	√
SEG20	F0414H	SEG20								√	√
SEG21	F0415H	SEG21								√	√
SEG22	F0416H	SEG22								√	√
SEG23	F0417H	SEG23								√	√
SEG24	F0418H	SEG24								√	–
SEG25	F0419H	SEG25								√	–
Use prohibited	F041AH	Use prohibited									
SEG26	F041BH	SEG26								√	–
SEG27	F041CH	SEG27								√	–
SEG28	F041DH	SEG28								√	–
SEG29	F041EH	SEG29								√	√
SEG30	F041FH	SEG30								√	–
SEG31	F0420H	SEG31								√	–
SEG32	F0421H	SEG32								√	–
SEG33	F0422H	SEG33								√	√
SEG34	F0423H	SEG34								√	√
SEG35	F0424H	SEG35								√	√
SEG36	F0425H	SEG36								√	√
SEG37	F0426H	SEG37								√	√
SEG38	F0427H	SEG38								√	√
SEG39	F0428H	SEG39								√	√

**Table 16-10. Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (4/4)****(b) 8-time slice (2/2)**

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	80-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0		
SEG40	F0429H	SEG40								√	√
SEG41	F042AH	SEG41								√	√
SEG42	F042BH	SEG42								√	√
SEG43	F042CH	SEG43								√	√
SEG44	F042DH	SEG44								√	√
SEG45	F042EH	SEG45								√	√
SEG46	F042FH	SEG46								√	√
SEG47	F0430H	SEG47								√	√
SEG48	F0431H	SEG48								√	√
SEG49	F0432H	SEG49								√	√
SEG50	F0433H	SEG50								√	√

**Note** The COM4 to COM7 pins and SEG0 to SEG3 pins are used alternatively.

**Remark** √: Supported, -: Not supported

To use the LCD display data register when the number of time slices is static, two, three, or four, the lower four bits and higher four bits of each address of the LCD display data register become an A-pattern area and a B-pattern area, respectively.

The correspondences between A-pattern area data and COM signals are as follows: bit 0 ⇔ COM0, bit 1 ⇔ COM1, bit 2 ⇔ COM2, and bit 3 ⇔ COM3.

The correspondences between B-pattern area data and COM signals are as follows: bit 4 ⇔ COM0, bit 5 ⇔ COM1, bit 6 ⇔ COM2, and bit 7 ⇔ COM3.

A-pattern area data will be displayed on the LCD panel when BLON = LCDSEL = 0 has been selected, and B-pattern area data will be displayed on the LCD panel when BLON = 0 and LCDSEL = 1 have been selected.

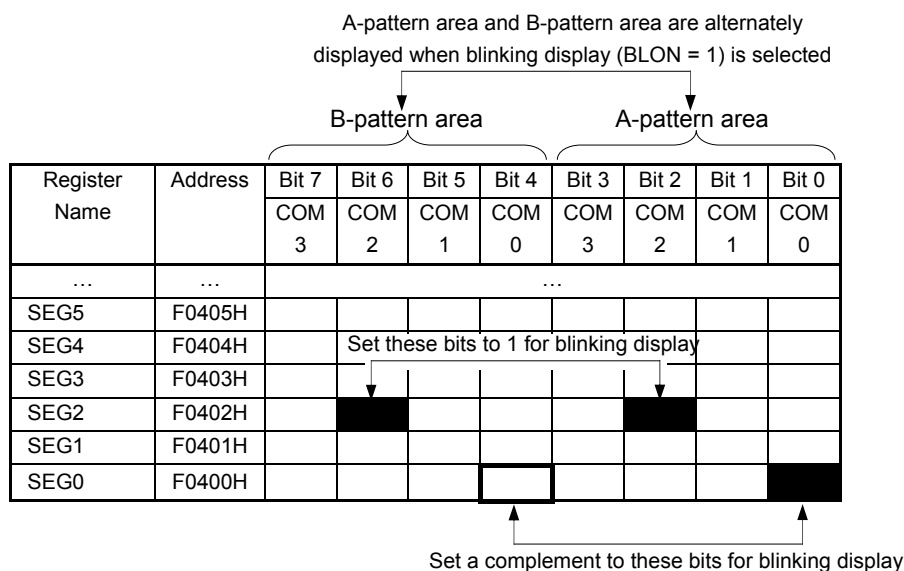
### 16.5 Selection of LCD Display Register

With RL78/L13, to use the LCD display data registers when the number of time slices is static, two, three, or four, the LCD display data register can be selected from the following three types, according to the BLON and LCDSEL bit settings.

- Displaying an A-pattern area data (lower four bits of LCD display data register)
- Displaying a B-pattern area data (higher four bits of LCD display data register)
- Alternately displaying A-pattern and B-pattern area data (blinking display corresponding to the constant-period interrupt timing of real-time clock 2 (RTC2))

**Caution** When the number of time slices is eight, LCD display data registers (A-pattern, B-pattern, or blinking display) cannot be selected.

**Figure 16-14. Example of Setting LCD Display Registers When Pattern Is Changed**



**16.5.1 A-pattern area and B-pattern area data display**

When BLON = LCDSEL = 0, A-pattern area (lower four bits of the LCD display data register) data will be output as the LCD display register.

When BLON = 0, and LCDSEL = 1, B-pattern area (higher four bits of the LCD display data register) data will be output as the LCD display register.

See **16.4 LCD Display Data Registers** for details about the display area.

**16.5.2 Blinking display (Alternately displaying A-pattern and B-pattern area data)**

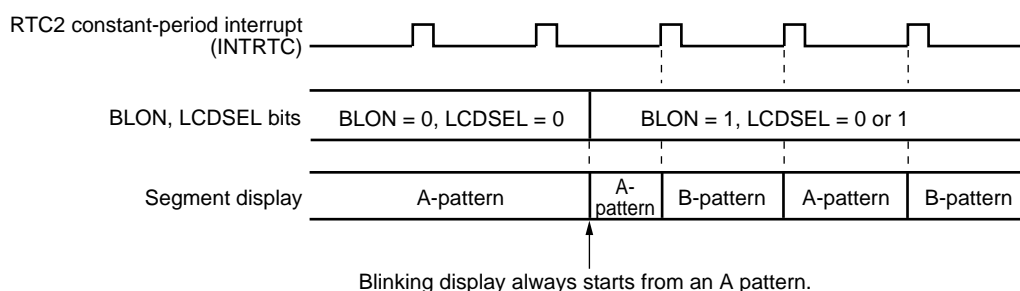
When BLON = 1 has been set, A-pattern and B-pattern area data will be alternately displayed, according to the constant-period interrupt (INTRTC) timing of real-time clock 2 (RTC2). See **CHAPTER 8 REAL-TIME CLOCK 2** about the setting of the RTC2 constant-period interrupt (INTRTC, 0.5 s setting only) timing.

For blinking display of the LCD, set inverted values to the B-pattern area bits corresponding to the A-pattern area bits. (Example: Write 1 to bit 0 of 00H, and set 0 to bit 4 of F0400H for blinking display.) When not setting blinking display of the LCD, set the same values. (Example: Write 1 to bit 2 of F0402H, and write 1 to bit 6 of F0402H for lighting display.)

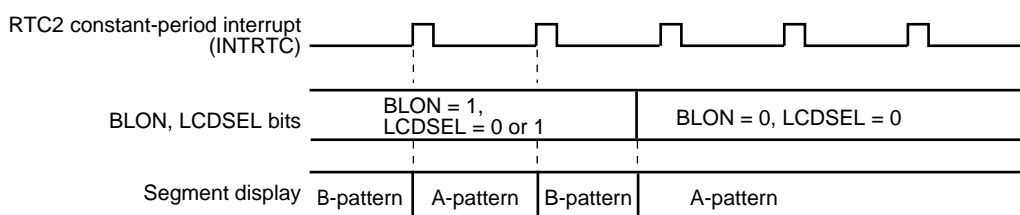
See **16.4 LCD Display Data Registers** for details about the display area.

Next, the timing operation of display switching is shown.

**Figure 16-15. Switching Operation from A-Pattern Display to Blinking Display**



**Figure 16-16. Switching Operation from Blinking Display to A-Pattern Display**



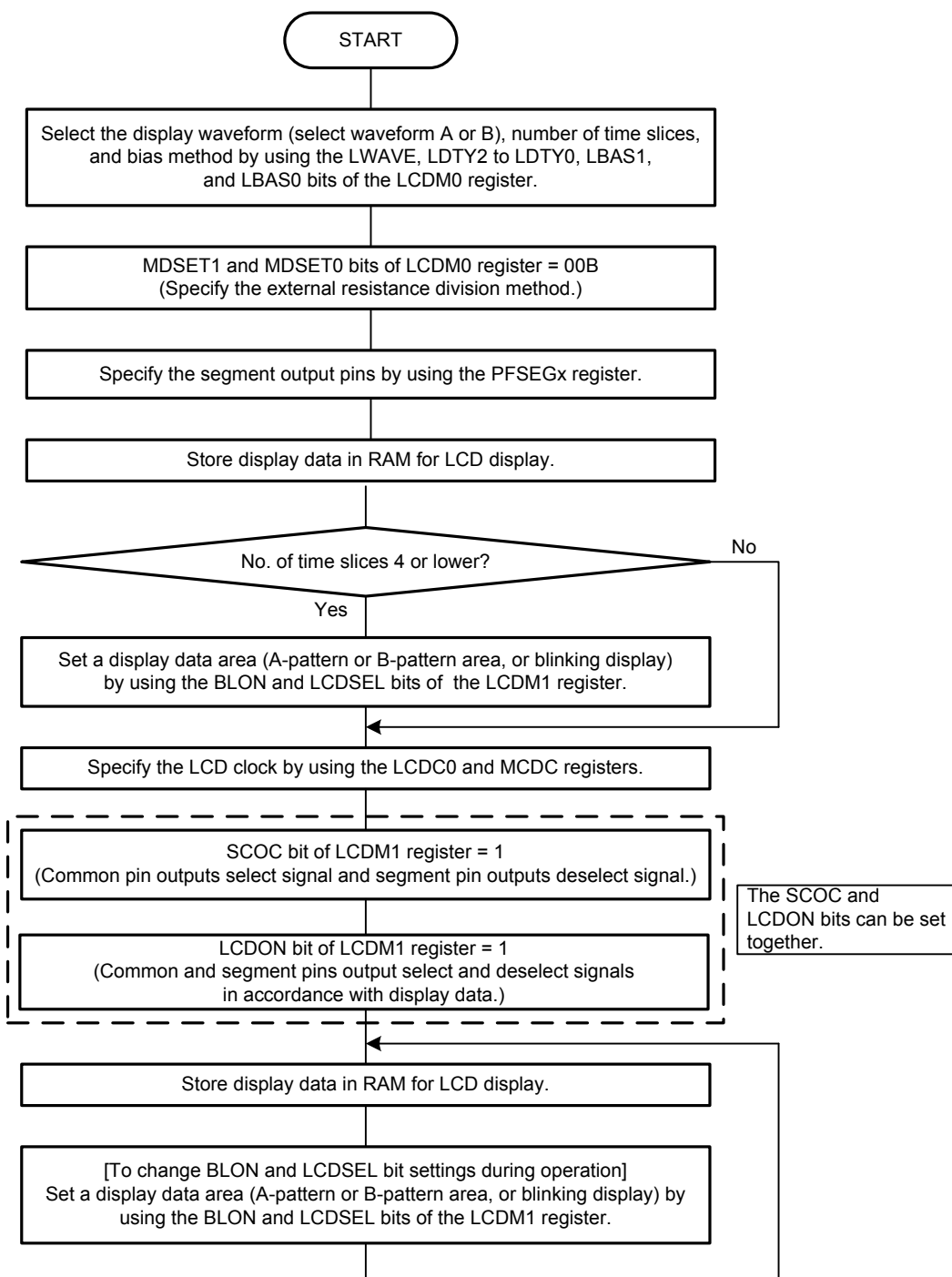
### 16.6 Setting the LCD Controller/Driver

Set the LCD controller/driver using the following procedure.

- Cautions**
1. To operate the LCD controller/driver, be sure to follow procedures (1) to (3). Unless these procedures are observed, the operation will not be guaranteed.
  2. The steps shown in the flowcharts in (1) to (3) are performed by the CPU.

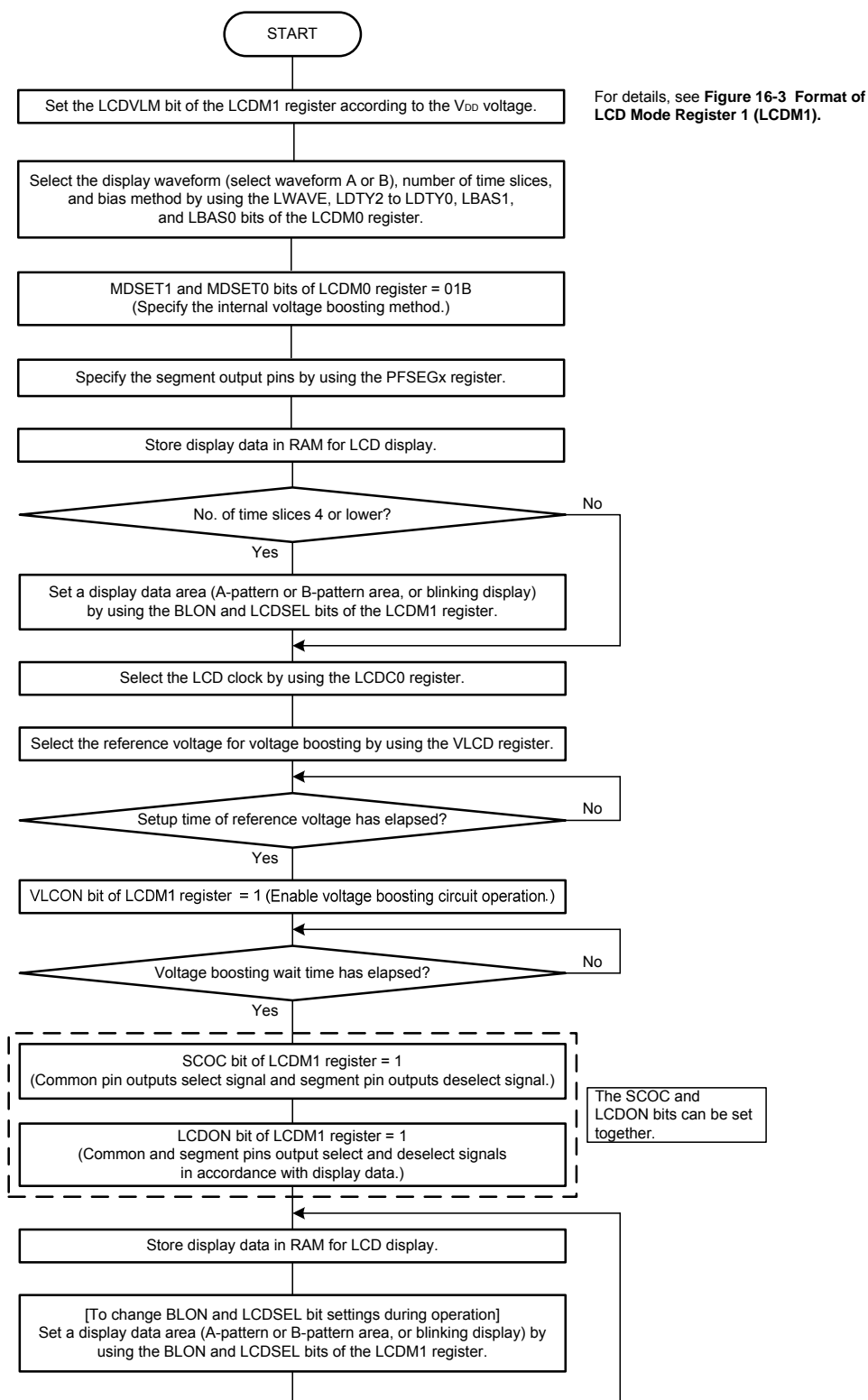
#### (1) External resistance division method

Figure 16-17. External Resistance Division Method Setting Procedure



(2) Internal voltage boosting method

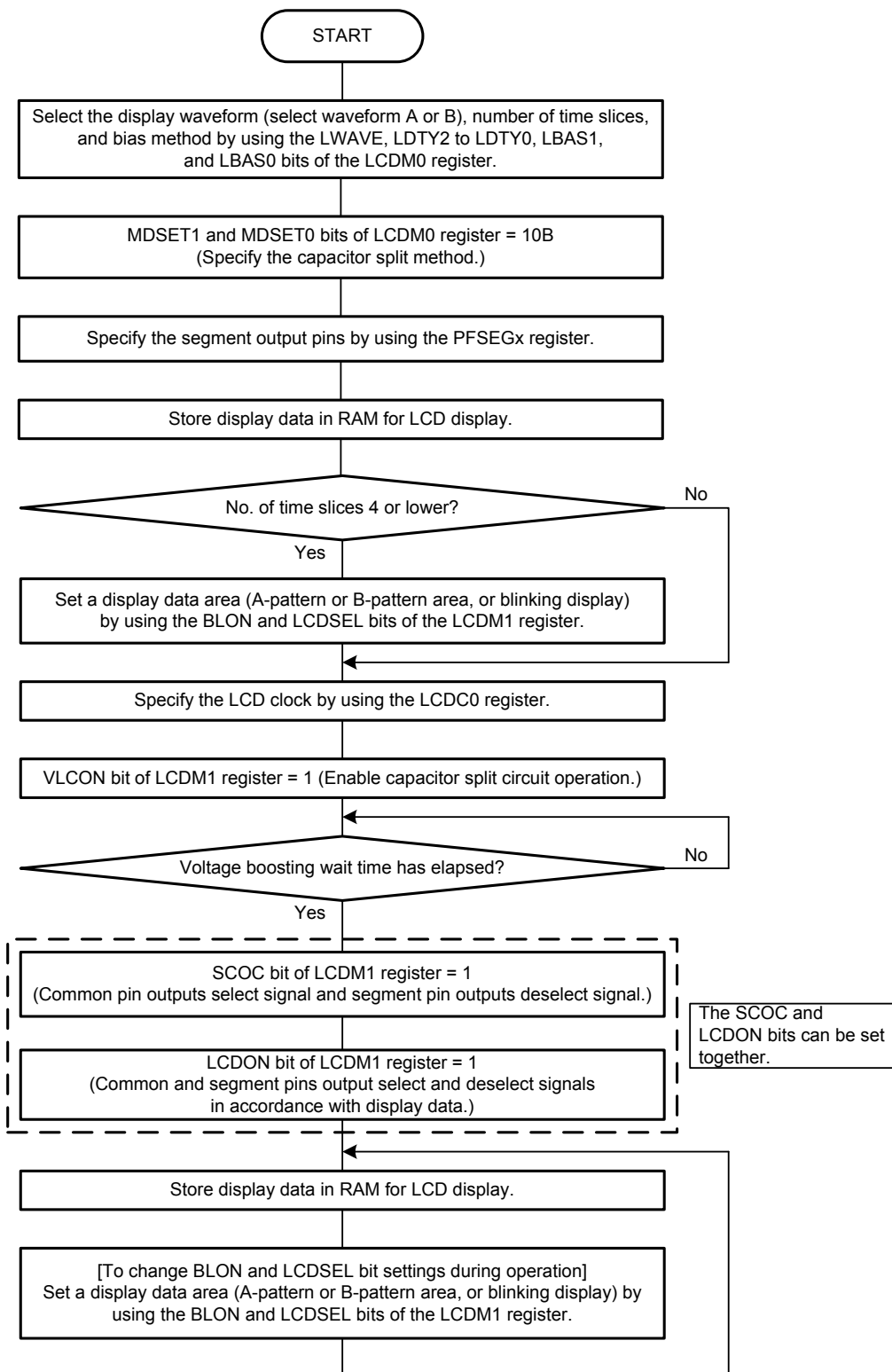
Figure 16-18. Internal Voltage Boosting Method Setting Procedure



- Cautions**
1. Wait until the setup time has elapsed even if not changing the setting of the VLCD register.
  2. For the specifications of the reference voltage setup time and voltage boosting wait time, see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)**.

(3) Capacitor split method

Figure 16-19. Capacitor Split Method Setting Procedure

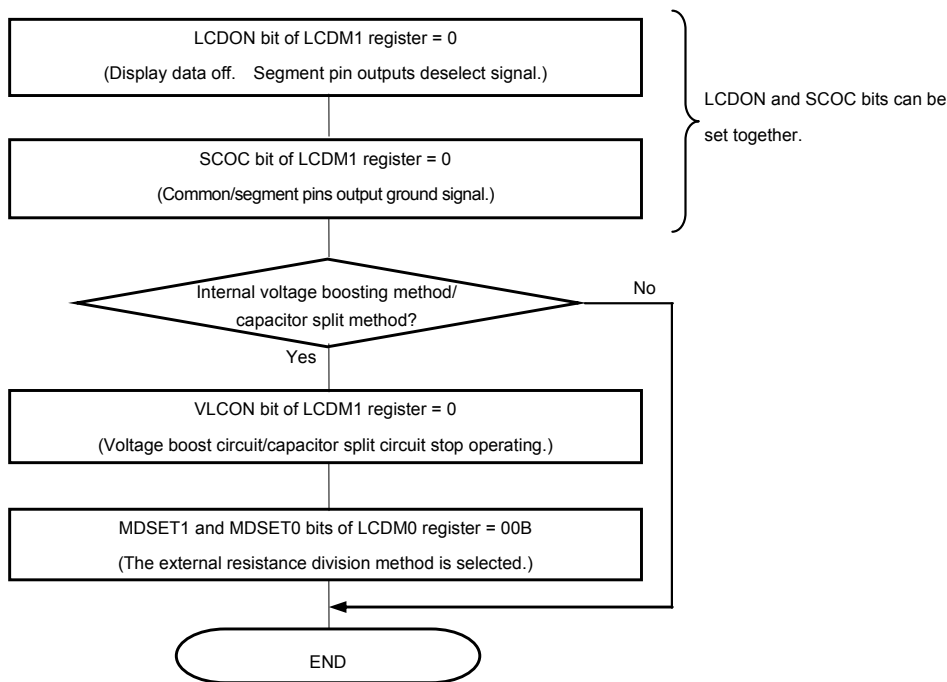


**Caution** For the specifications of the voltage boosting wait time, see CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C) and CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C).

### 16.7 Operation Stop Procedure

To stop the operation of the LCD while it is displaying waveforms, follow the steps shown in the flowchart below. The LCD stops operating when the LCDON bit of LCDM1 register and SCOC bit of the LCDM1 register are set to "0".

**Figure 16-20. Operation Stop Procedure**



**Caution** Stopping the voltage boost/capacitor split circuits is prohibited while the display is on (SCOC and LCDON bits of LCDM1 register = 11B). Otherwise, the operation will not be guaranteed. Be sure to turn off display (SCOC and LCDON bits of LCDM1 register = 10B) before stopping the voltage boost/capacitor split circuits (VLCON bit of LCDM1 register = 0).

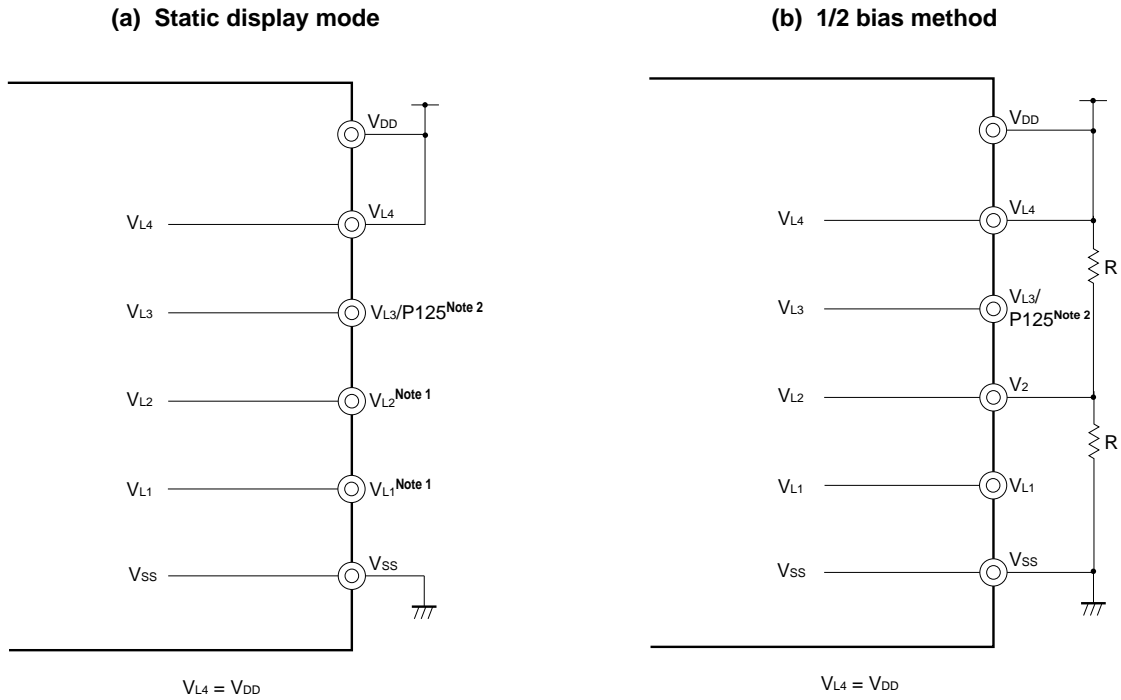


16.8 Supplying LCD Drive Voltages  $V_{L1}$ ,  $V_{L2}$ ,  $V_{L3}$ , and  $V_{L4}$

16.8.1 External resistance division method

Figure 16-21 shows examples of LCD drive voltage connection, corresponding to each bias method.

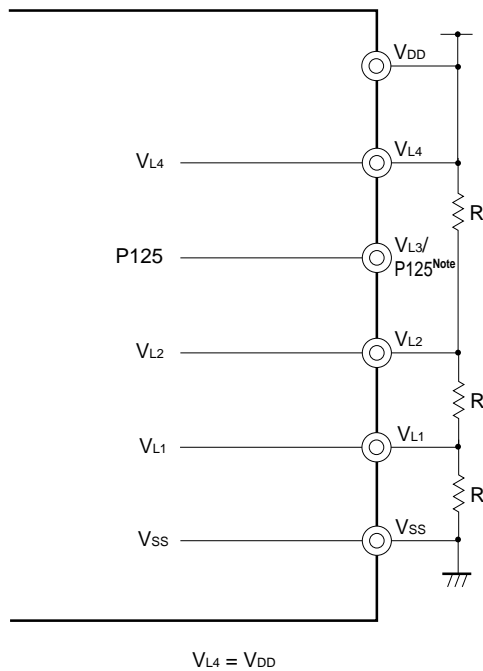
Figure 16-21. Examples of LCD Drive Power Connections (External Resistance Division Method) (1/2)



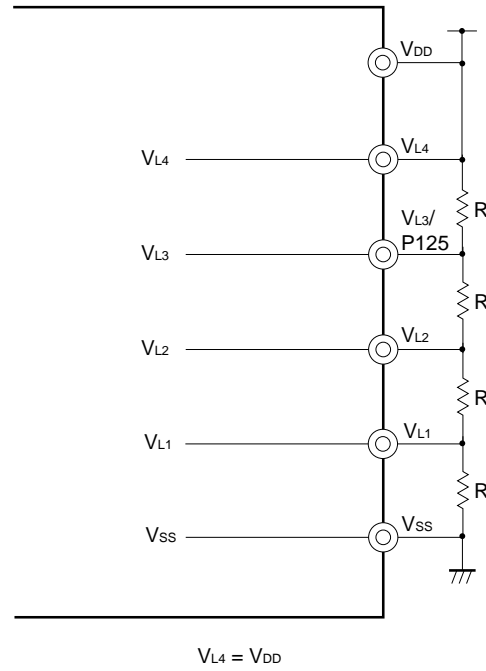
- Notes 1. Connect  $V_{L1}$  and  $V_{L2}$  to GND or leave open.
- 2.  $V_{L3}$  can be used as port (P125).

Figure 16-21. Examples of LCD Drive Power Connections (External Resistance Division Method) (2/2)

(c) 1/3 bias method



(d) 1/4 bias method



**Note** VL3 can be used as port (P125).

**Caution** The reference resistance “R” value for external resistance division is 10 kΩ to 1 MΩ. In addition, to stabilize the potential of the VL1 to VL4 pins, connect a capacitor between each of pins VL1 to VL4 and the GND pin as needed. The reference capacitance is about 0.22 μF but it depends on the LCD panel used, the number of segment pins, the number of common pins, the frame frequency, and the operating environment. Thoroughly evaluate these values in accordance with your system and adjust and determine the capacitance.

**16.8.2 Internal voltage boosting method**

The RL78/L13 contains an internal voltage boost circuit for generating LCD drive power supplies. The internal voltage boost circuit and external capacitors ( $0.47 \mu\text{F} \pm 30\%$ ) are used to generate an LCD drive voltage. Only 1/3 bias mode or 1/4 bias mode can be set for the internal voltage boosting method.

The LCD drive voltage of the internal voltage boosting method can supply a constant voltage, regardless of changes in  $V_{DD}$ , because it is a power supply separate from the main unit.

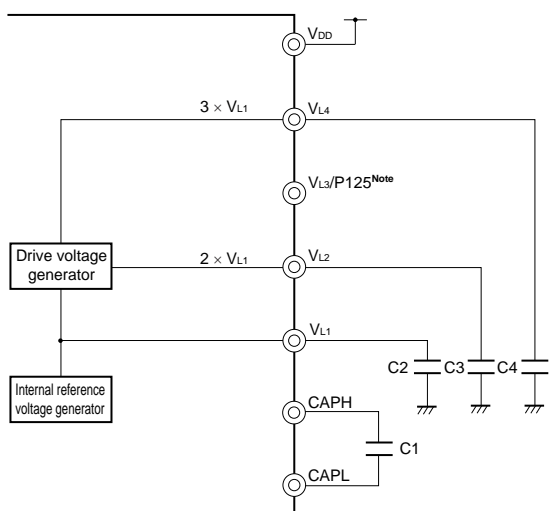
In addition, a contrast can be adjusted by using the LCD boost level control register (VLCD).

**Table 16-11. LCD Drive Voltages (Internal Voltage Boosting Method)**

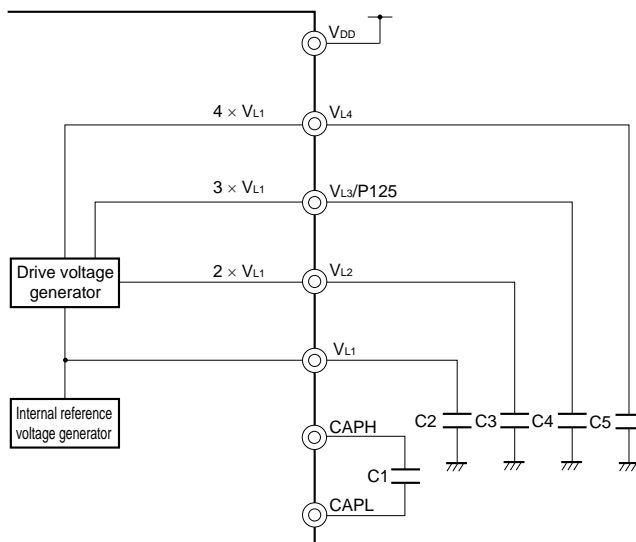
Bias Method \ LCD Drive Voltage Pin	1/3 Bias Method	1/4 Bias Method
$V_{L4}$	$3 \times V_{L1}$	$4 \times V_{L1}$
$V_{L3}$	—	$3 \times V_{L1}$
$V_{L2}$	$2 \times V_{L1}$	$2 \times V_{L1}$
$V_{L1}$	LCD reference voltage	LCD reference voltage

**Figure 16-22. Examples of LCD Drive Power Connections (Internal Voltage Boosting Method)**

**(a) 1/3 bias method**



**(b) 1/4 bias method**



**Note**  $V_{L3}$  can be used as port (P125).

**Remark** Use a capacitor with as little leakage as possible.  
In addition, make C1 a nonpolar capacitor.

**16.8.3 Capacitor split method**

The RL78/L13 contains an internal voltage reduction circuit for generating LCD drive power supplies. The internal voltage reduction circuit and external capacitors ( $0.47 \mu\text{F} \pm 30\%$ ) are used to generate an LCD drive voltage. Only 1/3 bias mode can be set for the capacitor split method.

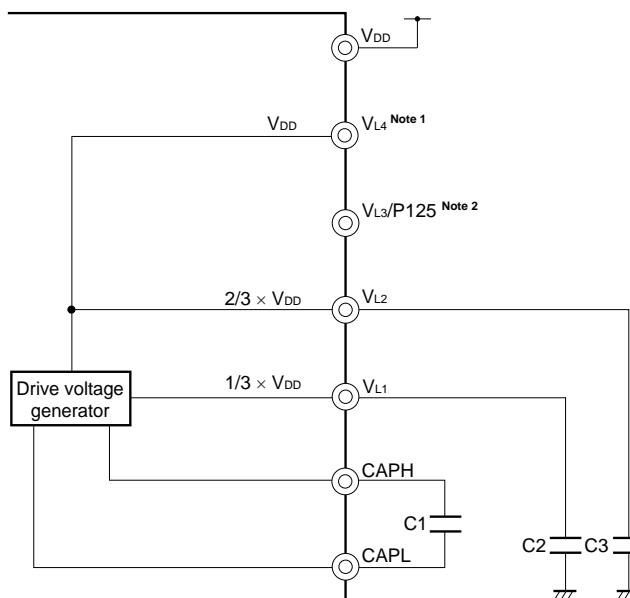
Different from the external resistance division method, there is always no current flowing with the capacitor split method, so current consumption can be reduced.

**Table 16-12. LCD Drive Voltages (Capacitor Split Method)**

Bias Method	1/3 Bias Method
LCD Drive Voltage Pin	
V <sub>L4</sub>	V <sub>DD</sub>
V <sub>L3</sub>	—
V <sub>L2</sub>	$\frac{2}{3} \times V_{L4}$
V <sub>L1</sub>	$\frac{1}{3} \times V_{L4}$

**Figure 16-23. Examples of LCD Drive Power Connections (Capacitor Split Method)**

**• 1/3 bias method**



**Notes 1.** When switching to internal voltage boosting method, connect capacitor C4 as shown in **Figure 16-22 Examples of LCD Drive Power Connections (Internal Voltage Boosting Method)**.

**2.** VL<sub>3</sub> can be used as port (P125).

**Remark** Use a capacitor with as little leakage as possible. In addition, make C1 a nonpolar capacitor.

### 16.9 Common and Segment Signals

Each pixel of the LCD panel turns on when the potential difference between the corresponding common and segment signals becomes higher than a specific voltage (LCD drive voltage,  $V_{LCD}$ ). The pixels turn off when the potential difference becomes lower than  $V_{LCD}$ .

Applying DC voltage to the common and segment signals of an LCD panel causes deterioration. To avoid this problem, this LCD panel is driven by AC voltage.

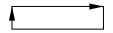
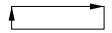
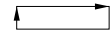
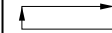
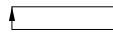
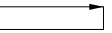
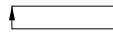
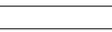

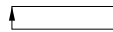
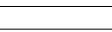
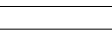
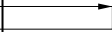








#### (1) Common signals

Each common signal is selected sequentially according to a specified number of time slices at the timing listed in Table 16-13. In the static display mode, the same signal is output to COM0 to COM3.

In the two-time-slice mode, leave the COM2 and COM3 pins open. In the three-time-slice mode, leave the COM3 pin open.

Use the COM4 to COM7 pins other than in the eight-time-slice mode as open or segment pins.

**Table 16-13. COM Signals**

COM Signal Number of Time Slices	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
Static display mode					Note	Note	Note	Note
Two-time-slice mode			Open	Open	Note	Note	Note	Note
Three-time-slice mode				Open	Note	Note	Note	Note
Four-time-slice mode					Note	Note	Note	Note
Eight-time-slice mode								

**Note** Use the pins as open or segment pins.

**(2) Segment signals**

The segment signals correspond to the LCD display data register (see **16.4 LCD Display Data Registers**).

When the number of time slices is eight, bits 0 to 7 of each display data register are read in synchronization with COM0 to COM7, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG4 to SEG50).

When the number of time slices is number other than eight, bits 0 to 3 of each byte in A-pattern area are read in synchronization with COM0 to COM3, and bits 4 to 7 of each byte in B-pattern area are read in synchronization with COM0 to COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG0 to SEG50).

Check, with the information given above, what combination of front-surface electrodes (corresponding to the segment signals) and rear-surface electrodes (corresponding to the common signals) forms display patterns in the LCD display data register, and write the bit data that corresponds to the desired display pattern on a one-to-one basis.

**Remark** The mounted segment output pins vary depending on the product.

- 64-pin products: SEG0 to SEG3, SEG6 to SEG8, SEG11, SEG12, SEG16 to SEG23, SEG29, SEG33 to SEG50
- 80-pin products: SEG0 to SEG50

**(3) Output waveforms of common and segment signals**

The voltages listed in Table 16-14 are output as common and segment signals.

When both common and segment signals are at the select voltage, a display on-voltage of  $\pm V_{LCD}$  is obtained. The other combinations of the signals correspond to the display off-voltage.

**Table 16-14. LCD Drive Voltage****(a) Static display mode**

Segment Signal		Select Signal Level	Deselect Signal Level
		$V_{SS}/V_{L4}$	$V_{L4}/V_{SS}$
Common Signal			
	$V_{L4}/V_{SS}$	$-V_{LCD}/+V_{LCD}$	0 V/0 V

**(b) 1/2 bias method**

Segment Signal		Select Signal Level	Deselect Signal Level
		$V_{SS}/V_{L4}$	$V_{L4}/V_{SS}$
Common Signal			
Select signal level	$V_{L4}/V_{SS}$	$-V_{LCD}/+V_{LCD}$	0 V/0 V
Deselect signal level	$V_{L2}$	$-\frac{1}{2}V_{LCD}/+\frac{1}{2}V_{LCD}$	$+\frac{1}{2}V_{LCD}/-\frac{1}{2}V_{LCD}$

**(c) 1/3 bias method (waveform A or B)**

Segment Signal		Select Signal Level	Deselect Signal Level
		$V_{SS}/V_{L4}$	$V_{L2}/V_{L1}$
Common Signal			
Select signal level	$V_{L4}/V_{SS}$	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$
Deselect signal level	$V_{L1}/V_{L2}$	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$	$+\frac{1}{3}V_{LCD}/-\frac{1}{3}V_{LCD}$

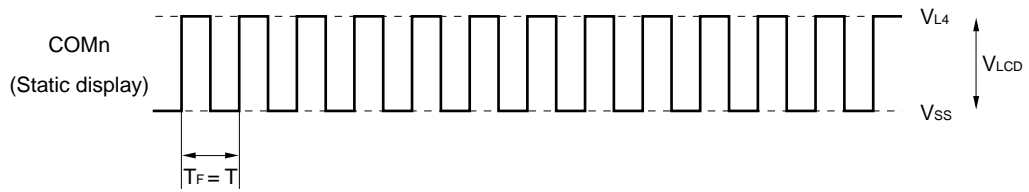
**(d) 1/4 bias method (waveform A or B)**

Segment Signal		Select Signal Level	Deselect Signal Level
		$V_{SS}/V_{L4}$	$V_{L2}$
Common Signal			
Select signal level	$V_{L4}/V_{SS}$	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{2}V_{LCD}/+\frac{1}{2}V_{LCD}$
Deselect signal level	$V_{L1}/V_{L3}$	$-\frac{1}{4}V_{LCD}/+\frac{1}{4}V_{LCD}$	$+\frac{1}{4}V_{LCD}/-\frac{1}{4}V_{LCD}$

Figure 16-24 shows the common signal waveforms, and Figure 16-25 shows the voltages and phases of the common and segment signals.

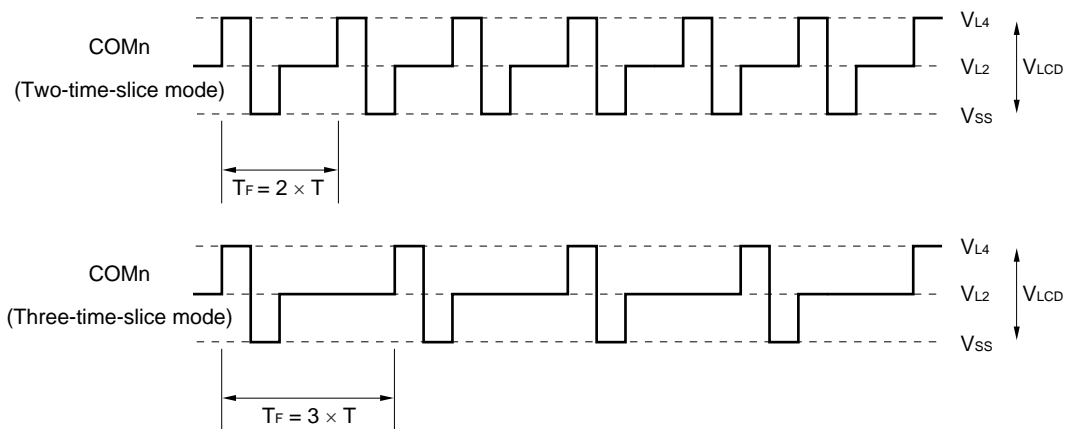
**Figure 16-24. Common Signal Waveforms (1/2)**

**(a) Static display mode**



T: One LCD clock period       $T_F$ : Frame frequency

**(b) 1/2 bias method**

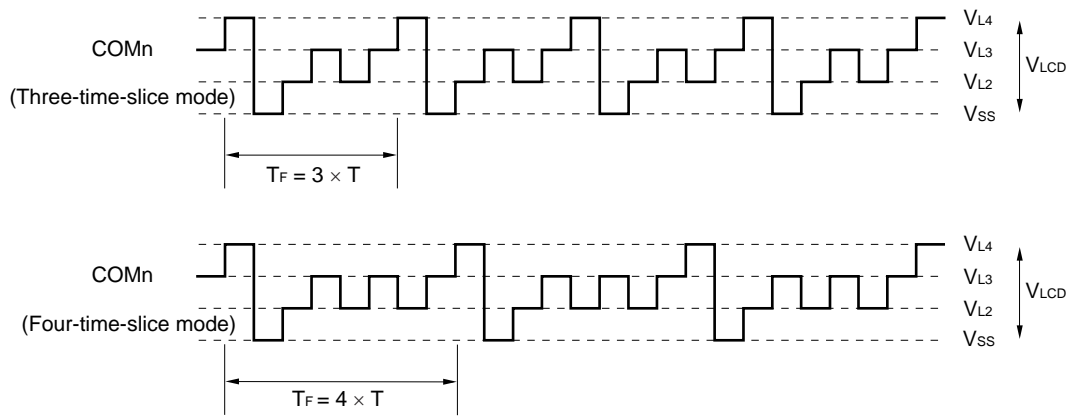


T: One LCD clock period       $T_F$ : Frame frequency



Figure 16-24. Common Signal Waveforms (2/2)

(c) 1/3 bias method



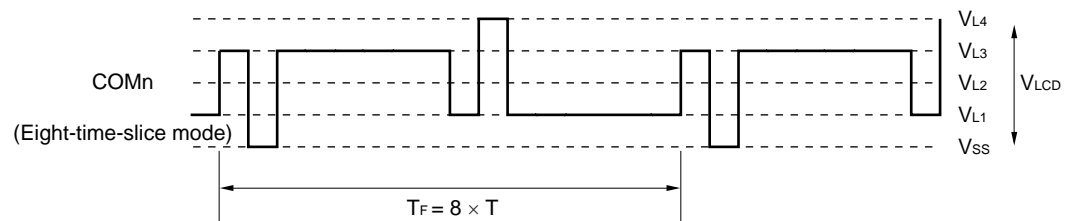
T: One LCD clock period       $T_F$ : Frame frequency

< Example of calculation of LCD frame frequency (When four-time slot mode is used) >

LCD clock:  $32768/2^7 = 256$  Hz (When setting to LCDC0 = 06H)

LCD frame frequency: 64 Hz

(d) 1/4 bias method



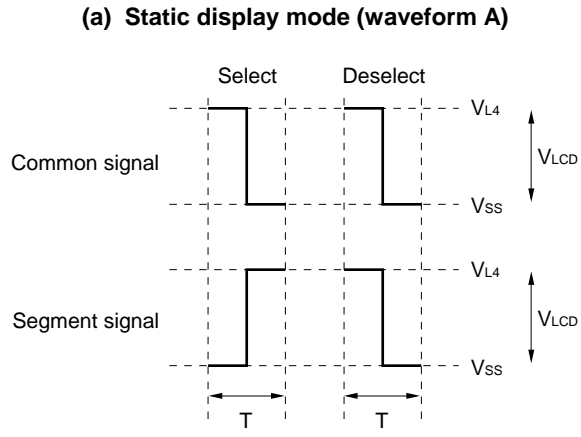
T: One LCD clock period       $T_F$ : Frame frequency

< Example of calculation of LCD frame frequency (When eight-time slot mode is used) >

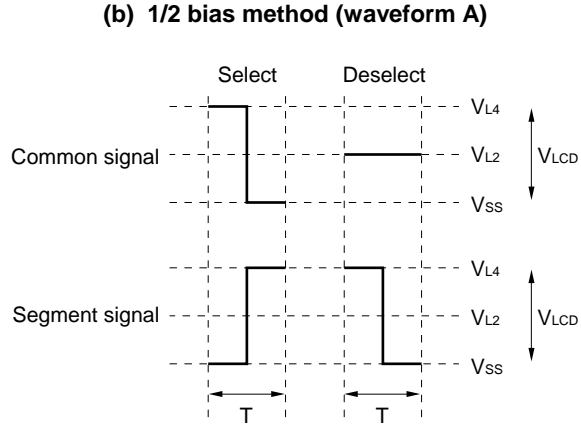
LCD clock:  $32768/2^7 = 256$  Hz (When setting to LCDC0 = 06H)

LCD frame frequency: 32 Hz

Figure 16-25. Voltages and Phases of Common and Segment Signals (1/3)



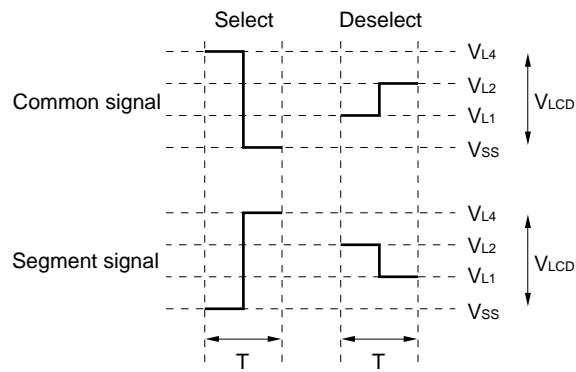
T: One LCD clock period



T: One LCD clock period

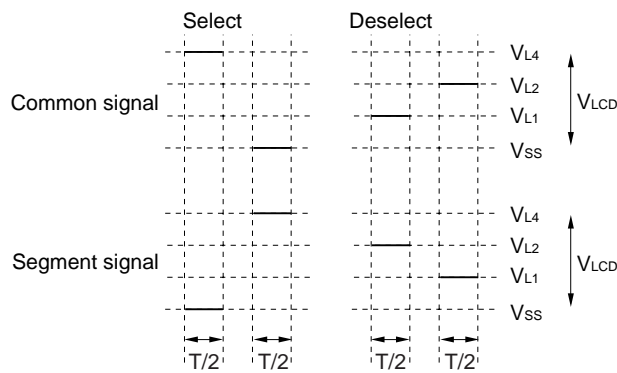
Figure 16-25. Voltages and Phases of Common and Segment Signals (2/3)

(c) 1/3 bias method (waveform A)



T: One LCD clock period

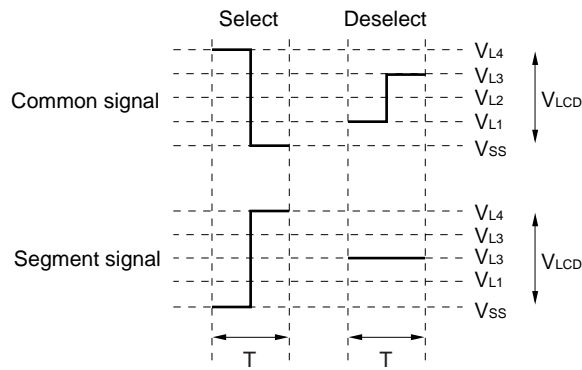
(d) 1/3 bias method (waveform B)



T: One LCD clock period

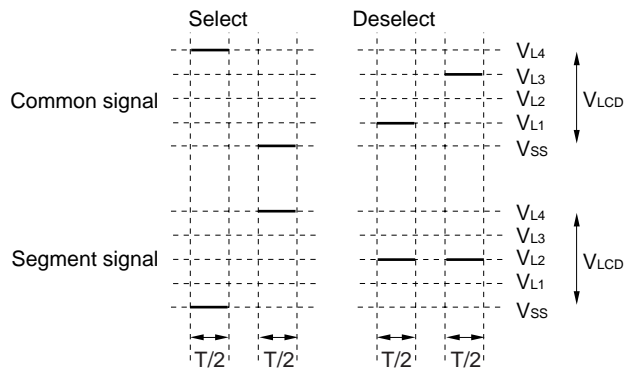
Figure 16-25. Voltages and Phases of Common and Segment Signals (3/3)

(e) 1/4 bias method (waveform A)



T: One LCD clock period

(f) 1/4 bias method (waveform B)



T: One LCD clock period

### 16.10 Display Modes

#### 16.10.1 Static display example

Figure 16-27 shows how the three-digit LCD panel having the display pattern shown in Figure 16-26 is connected to the segment signals (SEG0 to SEG23) and the common signal (COM0). This example displays data “12.3” in the LCD panel. The contents of the display data register (F0400H to F0417H) correspond to this display.

The following description focuses on numeral “2.” (2.) displayed in the second digit. To display “2.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG8 to SEG15 pins according to Table 16-15 at the timing of the common signal COM0; see Figure 16-26 for the relationship between the segment signals and LCD segments.

**Table 16-15. Select and Deselect Voltages (COM0)**

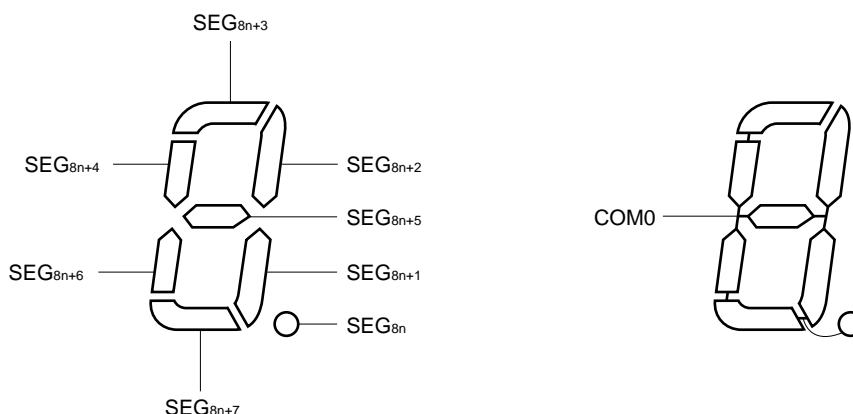
Segment	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
Common								
COM0	Select	Deselect	Select	Select	Deselect	Select	Select	Select

According to the above, it is determined that the bit-0 pattern of the display data register locations (F0408H to F040FH) must be 10110111.

Figure 16-28 shows the LCD drive waveforms of SEG11 and SEG12, and COM0. When the select voltage is applied to SEG11 at the timing of COM0, an alternate rectangle waveform, +V<sub>LCD</sub>/-V<sub>LCD</sub>, is generated to turn on the corresponding LCD segment.

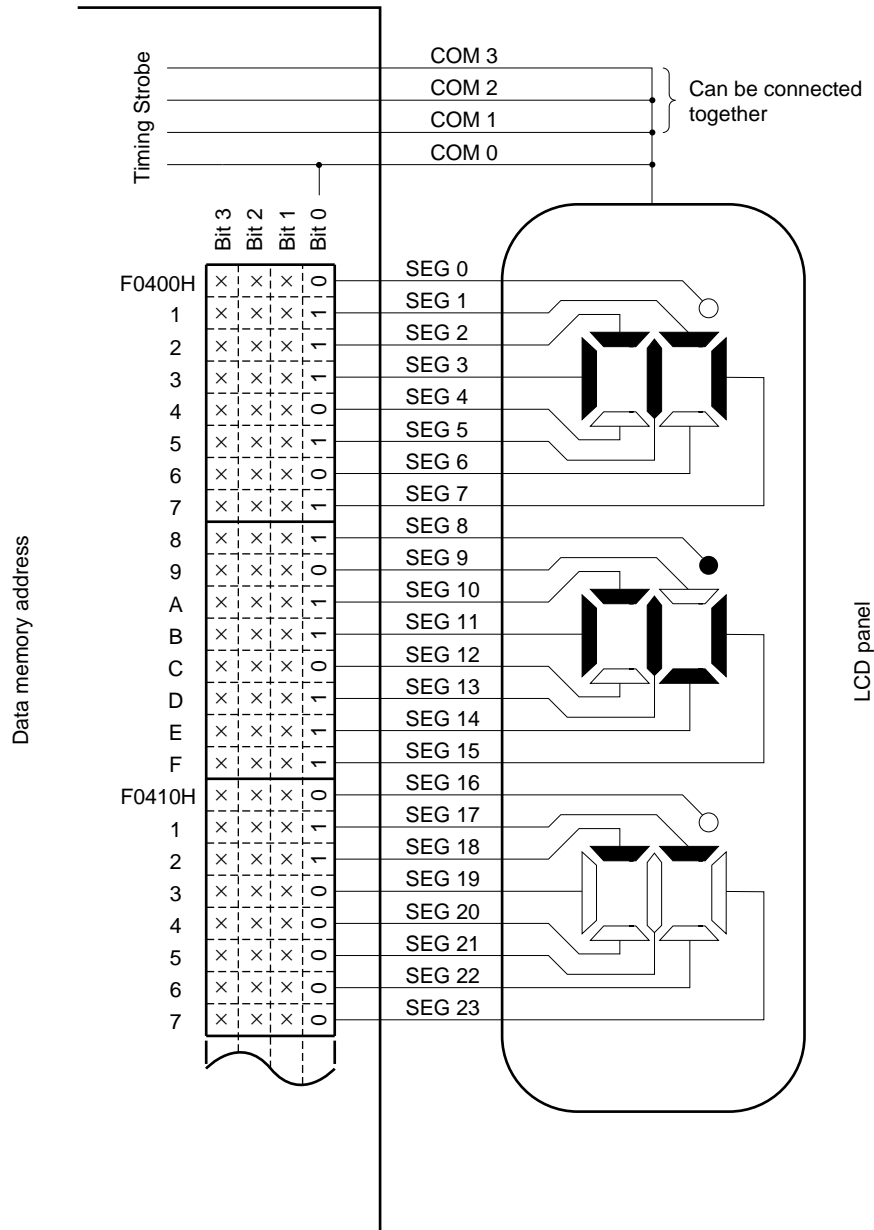
COM1 to COM3 are supplied with the same waveform as for COM0. So, COM0 to COM3 may be connected together to increase the driving capacity.

**Figure 16-26. Static LCD Display Pattern and Electrode Connections**

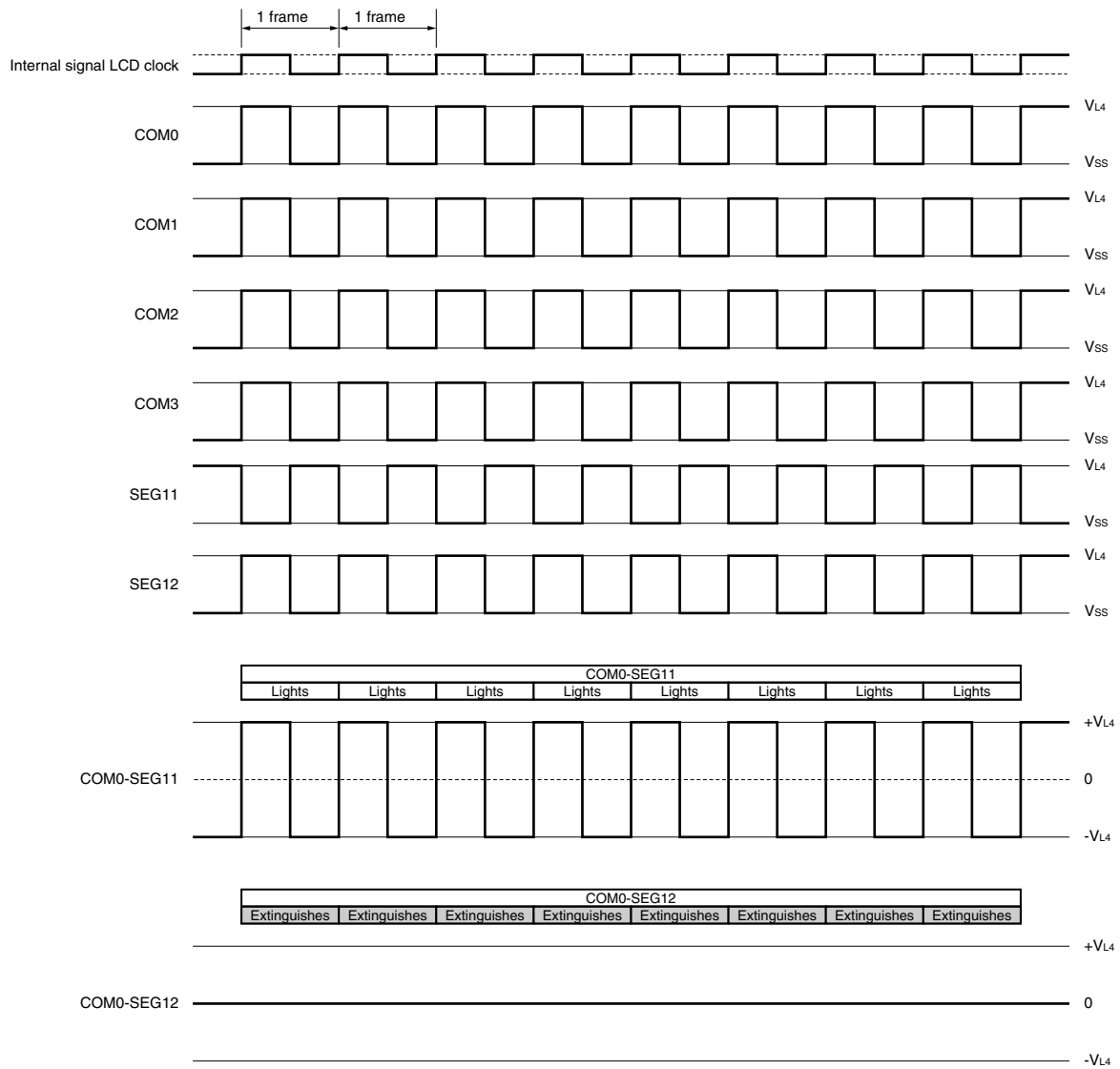


**Remark** 80-pin products: n = 0 to 5

Figure 16-27. Example of Connecting Static LCD Panel



**Figure 16-28. Static LCD Drive Waveform Examples for SEG11, SEG2, and COM0**



**16.10.2 Two-time-slice display example**

Figure 16-30 shows how the 6-digit LCD panel having the display pattern shown in Figure 16-29 is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 and COM1). This example displays data “12345.6” in the LCD panel. The contents of the display data register (F0400H to F0417H) correspond to this display.

The following description focuses on numeral “3” ( 3 ) displayed in the fourth digit. To display “3” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG12 to SEG15 pins according to Table 16-16 at the timing of the common signals COM0 and COM1; see Figure 16-29 for the relationship between the segment signals and LCD segments.

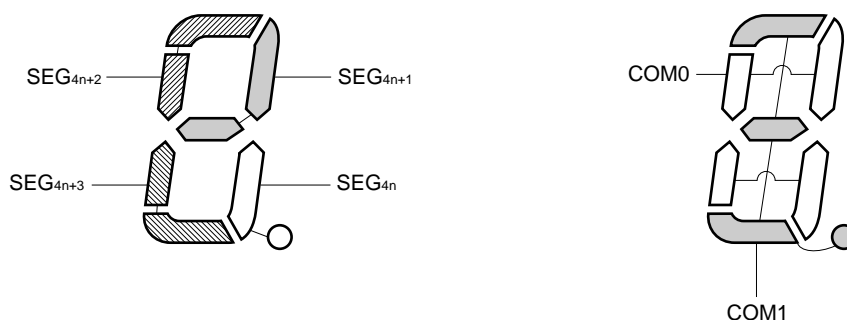
**Table 16-16. Select and Deselect Voltages (COM0 and COM1)**

Segment	SEG12	SEG13	SEG14	SEG15
Common				
COM0	Select	Select	Deselect	Deselect
COM1	Deselect	Select	Select	Select

According to the above, it is determined that the display data register location (F040FH) that corresponds to SEG15 must contain xx10.

Figure 16-31 shows examples of LCD drive waveforms between the SEG15 signal and each common signal. When the select voltage is applied to SEG15 at the timing of COM1, an alternate rectangle waveform, +V<sub>LCD</sub>/–V<sub>LCD</sub>, is generated to turn on the corresponding LCD segment.

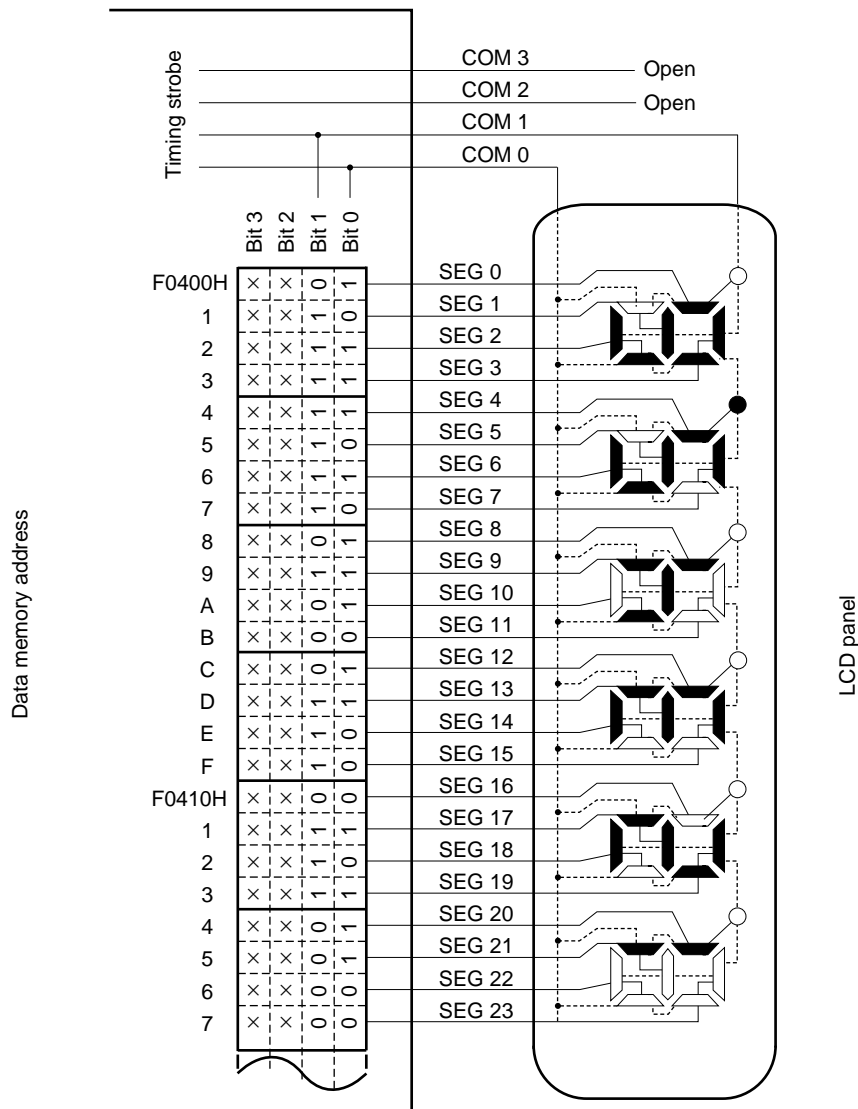
**Figure 16-29. Two-Time-Slice LCD Display Pattern and Electrode Connections**



**Remark** 80-pin products: n = 0 to 12

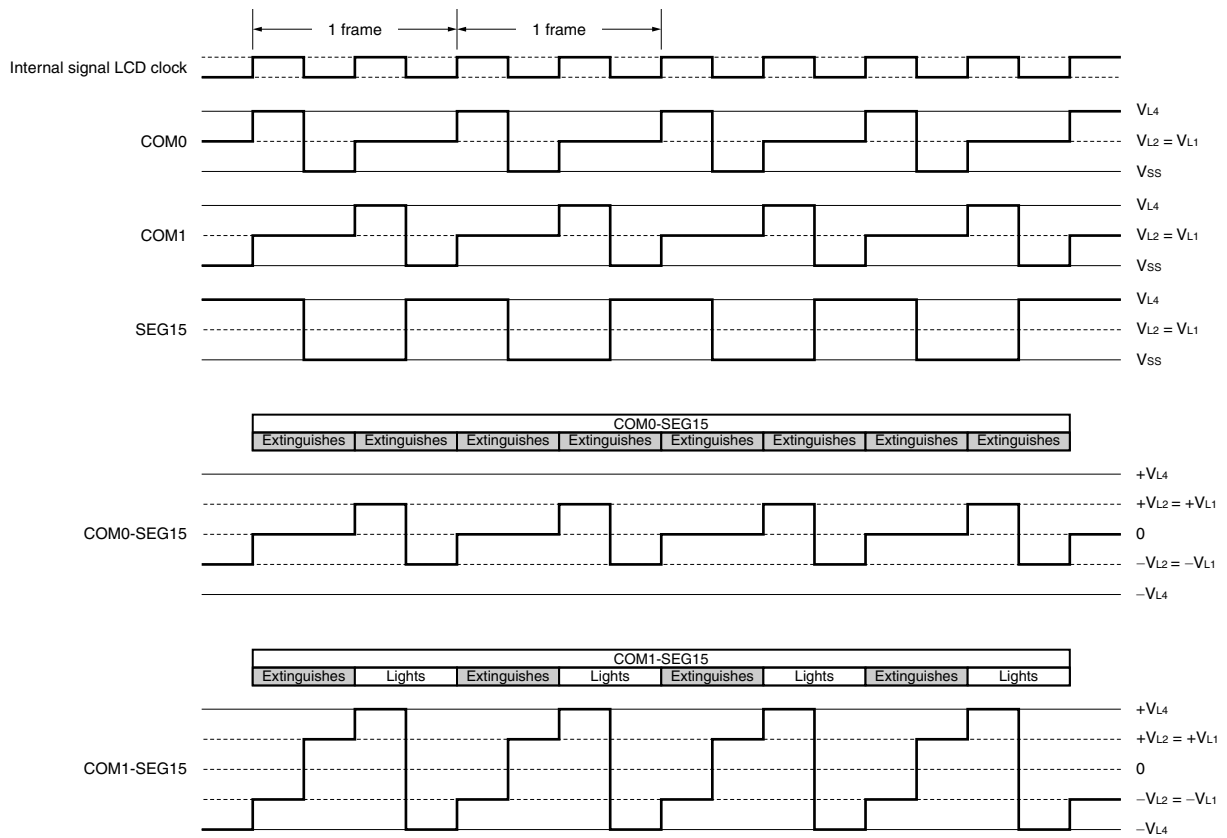


Figure 16-30. Example of Connecting Two-Time-Slice LCD Panel



x: Can always be used to store any data because the two-time-slice mode is being used.

**Figure 16-31. Two-Time-Slice LCD Drive Waveform Examples Between SEG15 and Each Common Signals (1/2 Bias Method)**



**16.10.3 Three-time-slice display example**

Figure 16-33 shows how the 8-digit LCD panel having the display pattern shown in Figure 16-32 is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 to COM2). This example displays data “123456.78” in the LCD panel. The contents of the display data register (addresses F0400H to F0417H) correspond to this display.

The following description focuses on numeral “6.” ( 6. ) displayed in the third digit. To display “6.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG6 to SEG8 pins according to Table 16-17 at the timing of the common signals COM0 to COM2; see Figure 16-32 for the relationship between the segment signals and LCD segments.

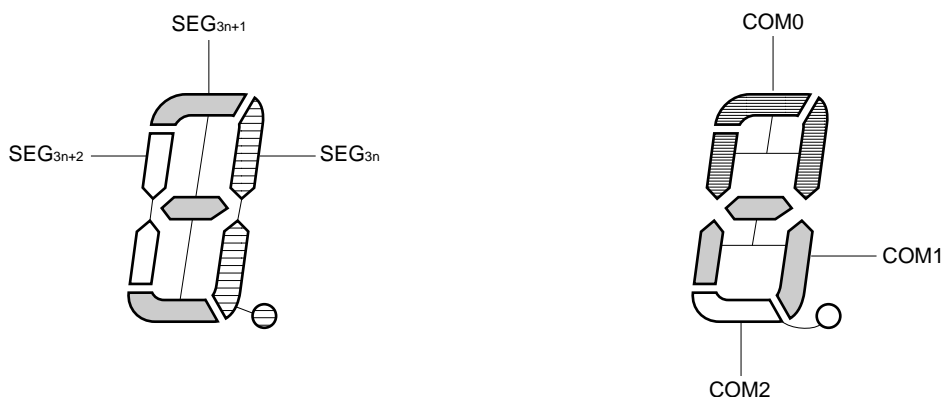
**Table 16-17. Select and Deselect Voltages (COM0 to COM2)**

Segment	SEG6	SEG7	SEG8
Common			
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	–

According to the above, it is determined that the display data register location (F0406H) that corresponds to SEG6 must contain x110.

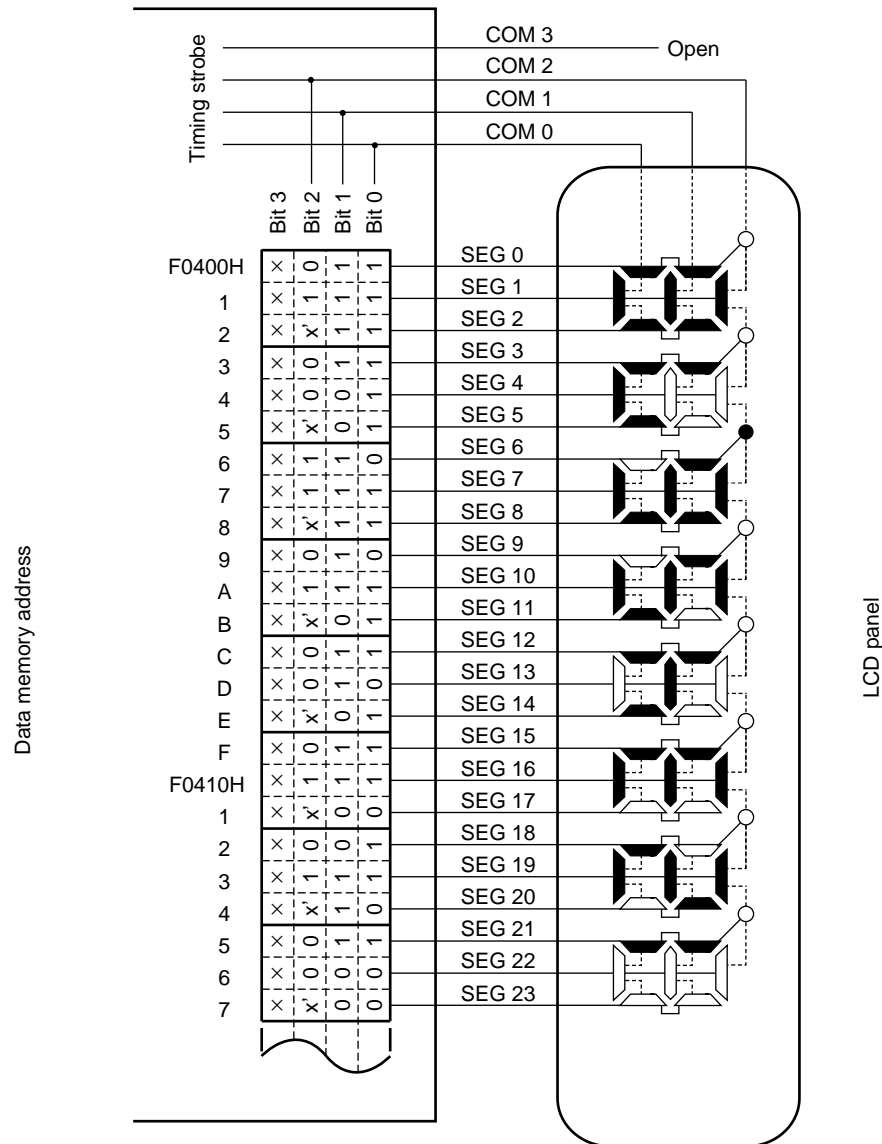
Figures 16-34 and 16-35 show examples of LCD drive waveforms between the SEG6 signal and each common signal in the 1/2 and 1/3 bias methods, respectively. When the select voltage is applied to SEG6 at the timing of COM1 or COM2, an alternate rectangle waveform, +V<sub>LCD</sub>/–V<sub>LCD</sub>, is generated to turn on the corresponding LCD segment.

**Figure 16-32. Three-Time-Slice LCD Display Pattern and Electrode Connections**



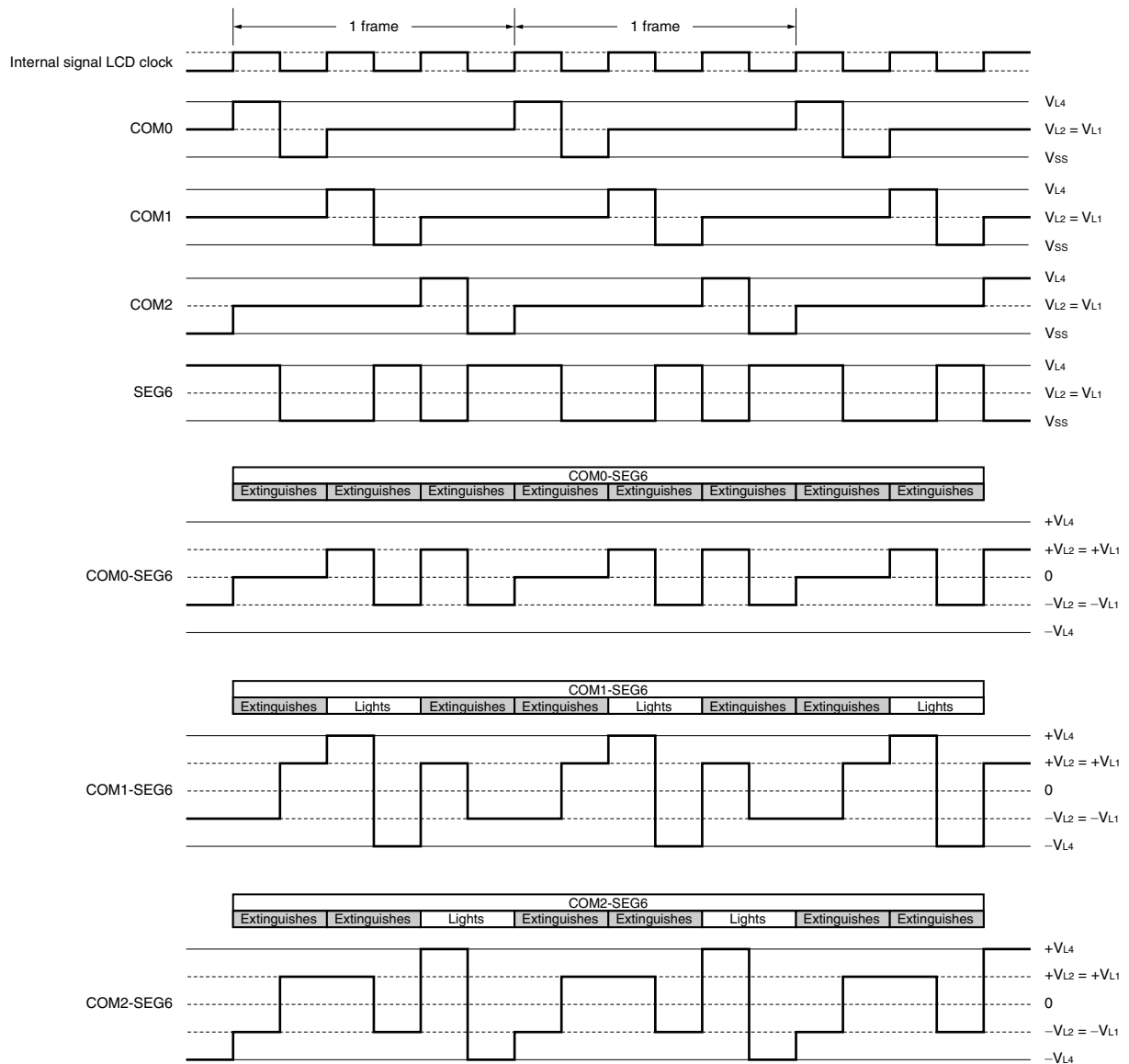
**Remark** 80-pin products: n = 0 to 16

Figure 16-33. Example of Connecting Three-Time-Slice LCD Panel

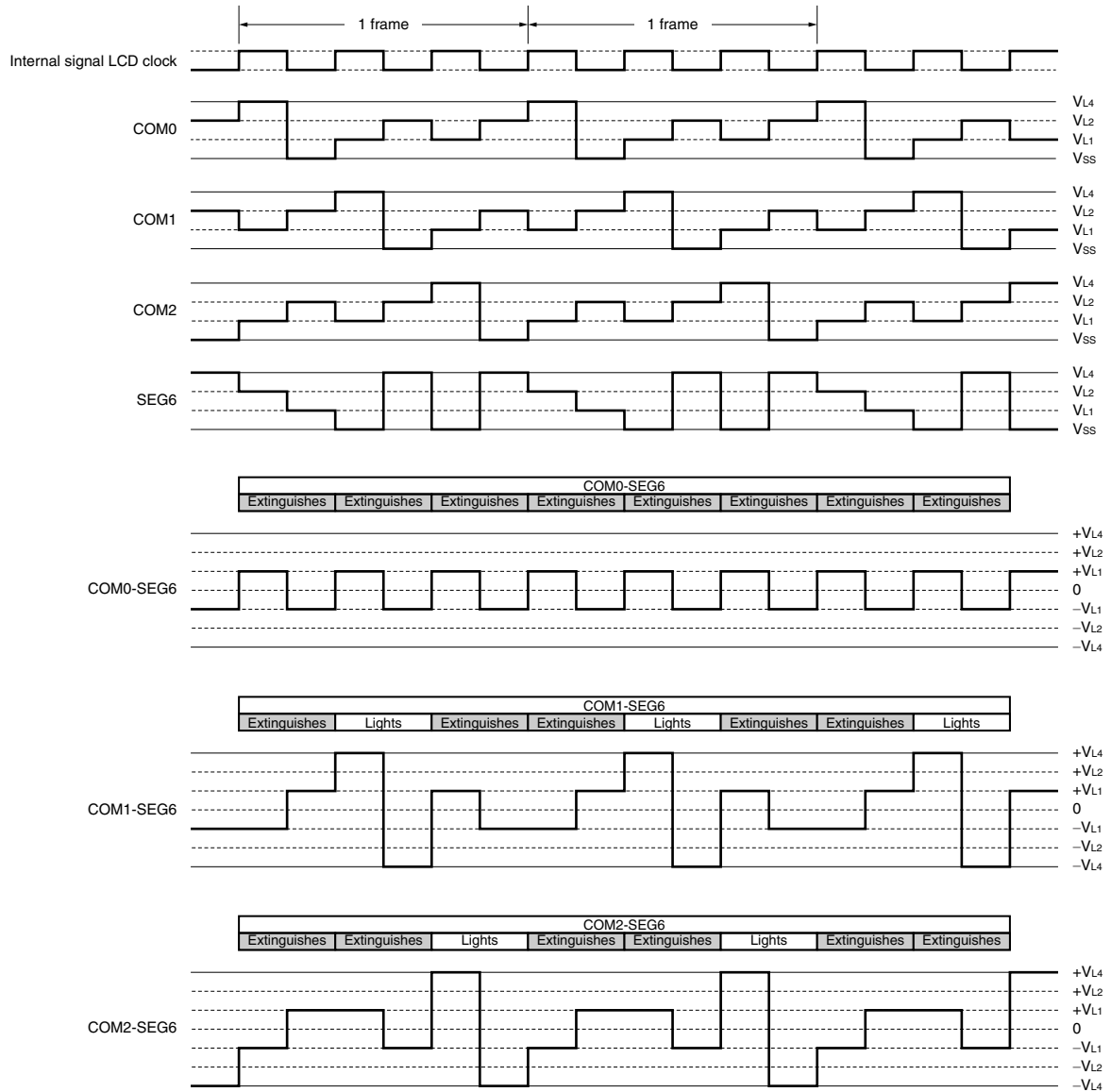


- x': Can be used to store any data because there is no corresponding segment in the LCD panel.
- x: Can always be used to store any data because the three-time-slice mode is being used.

**Figure 16-34. Three-Time-Slice LCD Drive Waveform Examples Between SEG6 and Each Common Signals (1/2 Bias Method)**



**Figure 16-35. Three-Time-Slice LCD Drive Waveform Examples Between SEG6 and Each Common Signals (1/3 Bias Method)**



**16.10.4 Four-time-slice display example**

Figure 16-37 shows how the 12-digit LCD panel having the display pattern shown in Figure 16-36 is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 to COM3). This example displays data “123456.789012” in the LCD panel. The contents of the display data register (addresses F0400H to F0417H) correspond to this display.

The following description focuses on numeral “6.” (6.) displayed in the seventh digit. To display “6.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG12 and SEG13 pins according to Figure 16-18 at the timing of the common signals COM0 to COM3; see Figure 16-36 for the relationship between the segment signals and LCD segments.

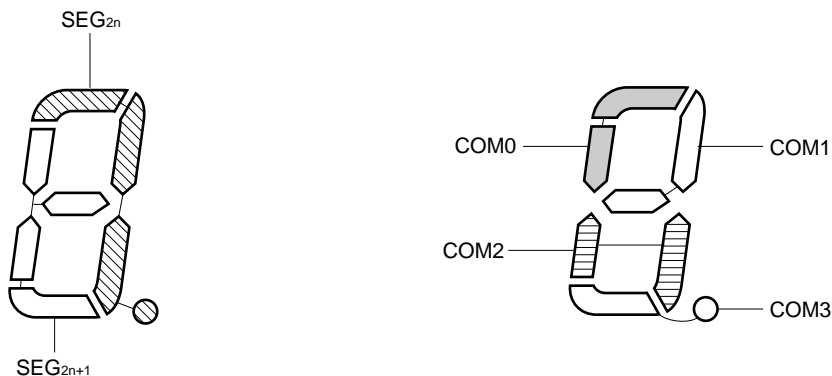
**Table 16-18. Select and Deselect Voltages (COM0 to COM3)**

Segment	SEG12	SEG13
Common		
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to the above, it is determined that the display data register location (F040CH) that corresponds to SEG12 must contain 1101.

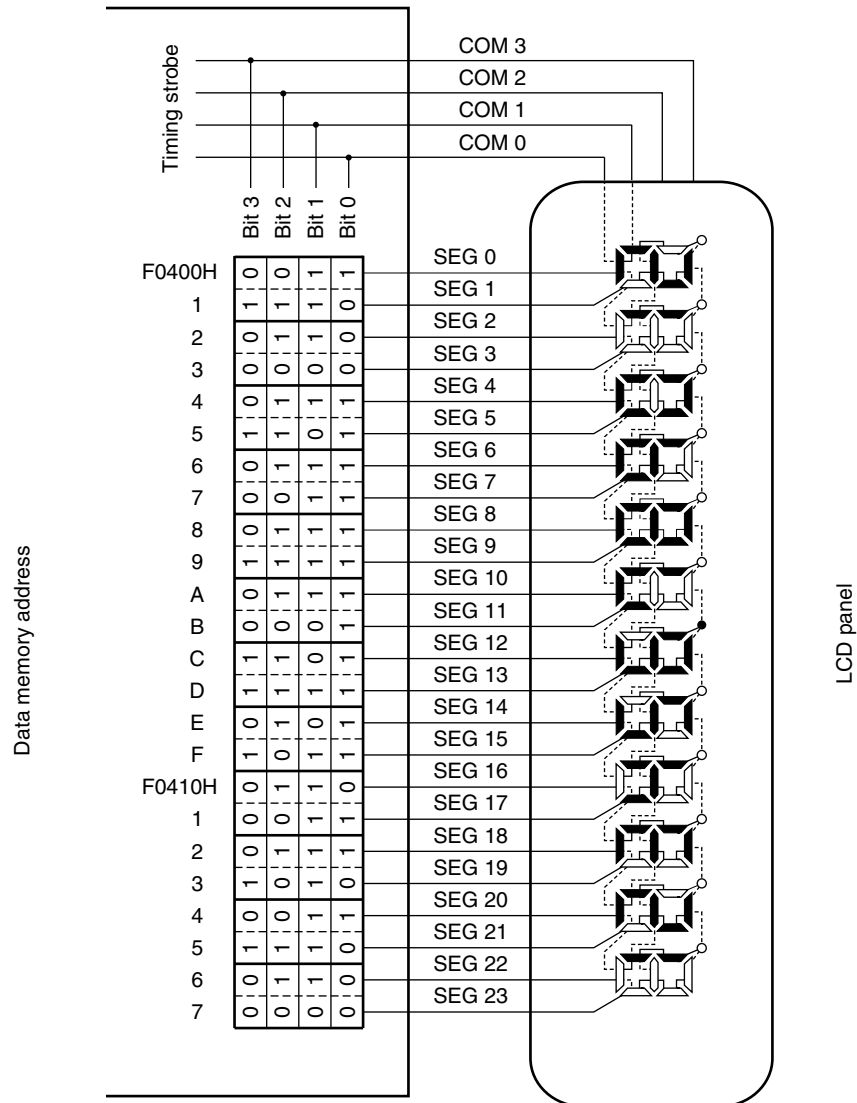
Figure 16-38 shows examples of LCD drive waveforms between the SEG12 signal and each common signal. When the select voltage is applied to SEG12 at the timing of COM0, an alternate rectangle waveform, +V<sub>LCD</sub>/–V<sub>LCD</sub>, is generated to turn on the corresponding LCD segment.

**Figure 16-36. Four-Time-Slice LCD Display Pattern and Electrode Connections**



**Remark** 80-pin products: n = 0 to 25

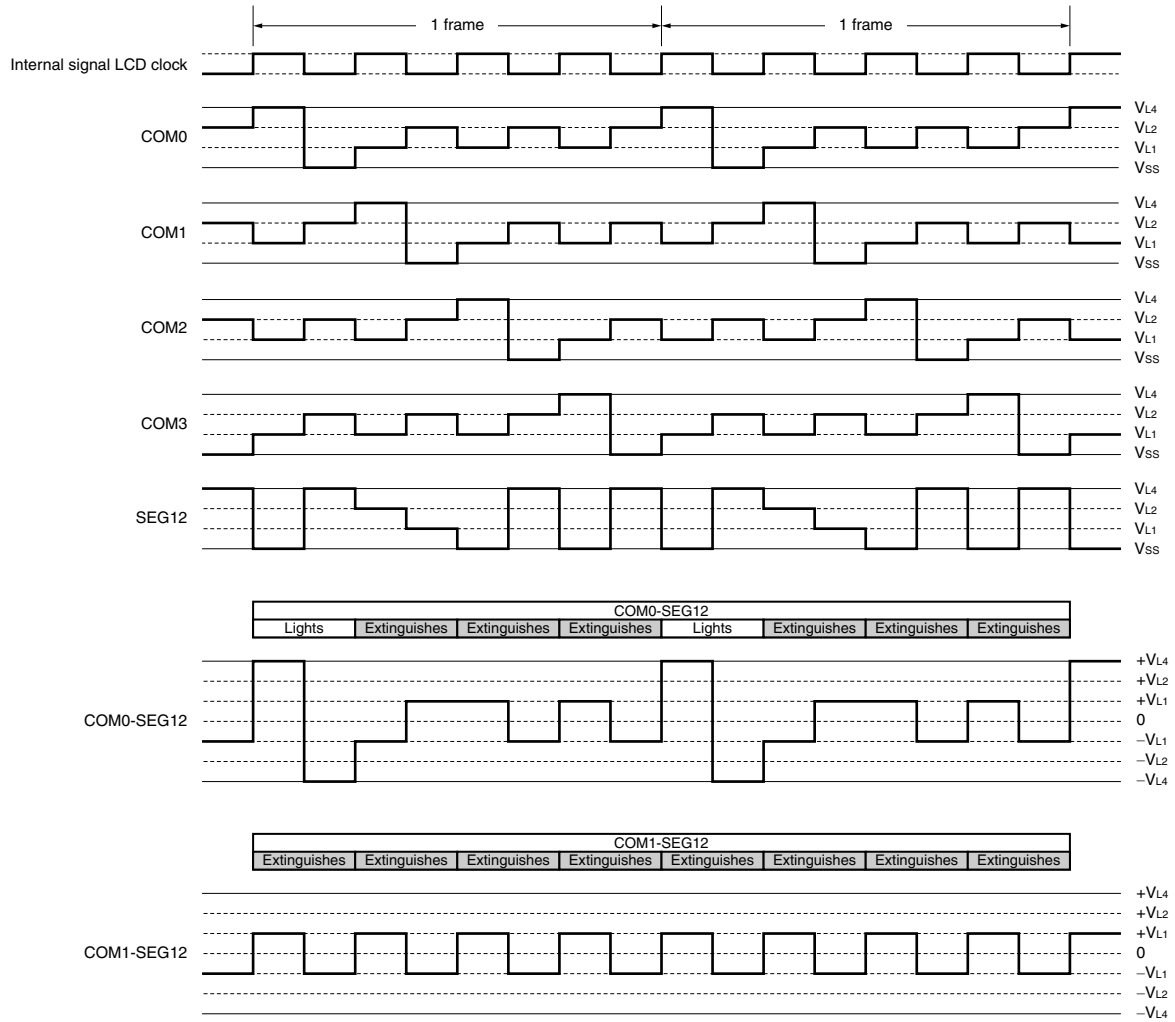
Figure 16-37. Example of Connecting Four-Time-Slice LCD Panel





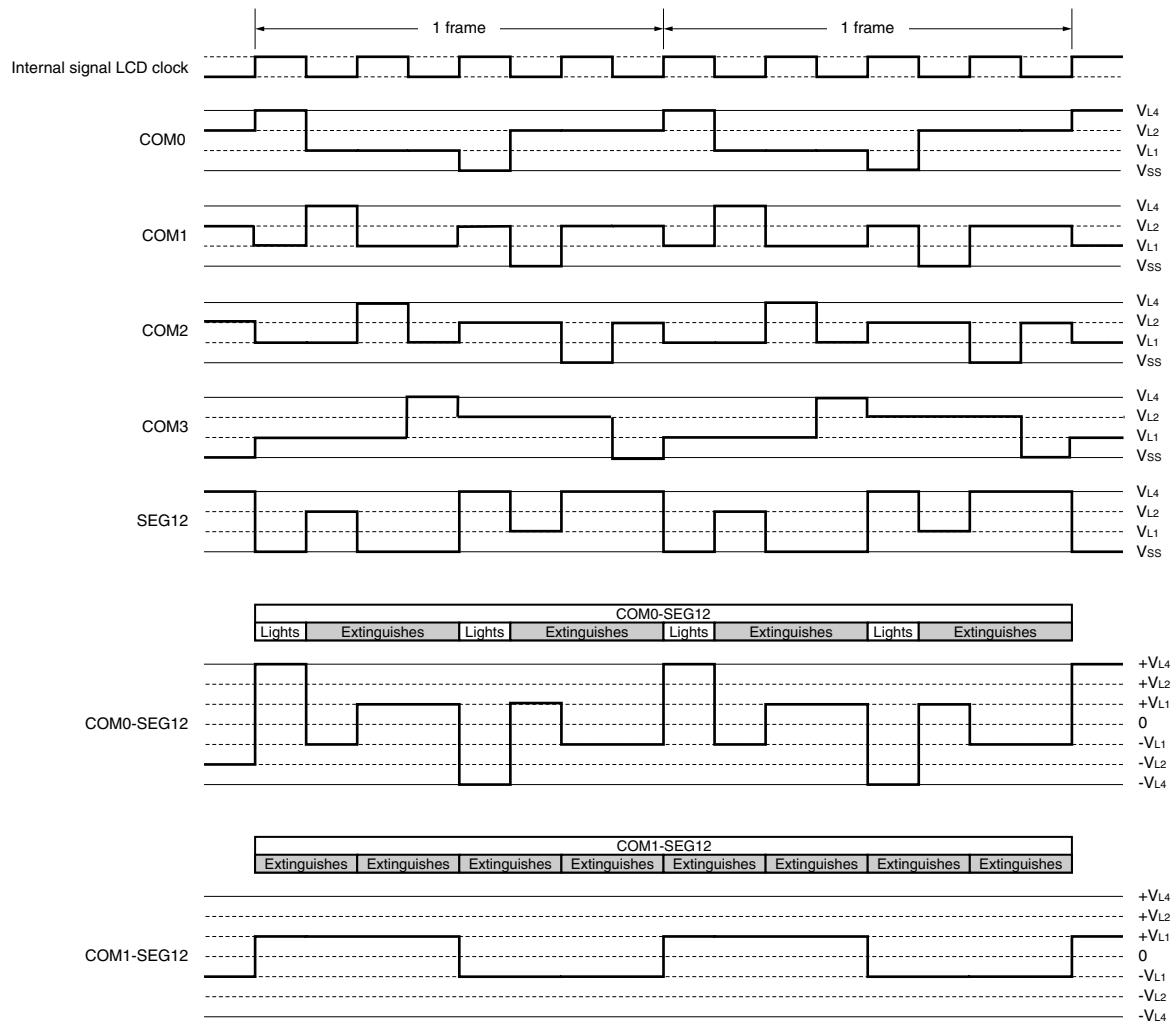
**Figure 16-38. Four-Time-Slice LCD Drive Waveform Examples Between SEG12 and Each Common Signals (1/3 Bias Method) (1/2)**

**(a) Waveform A**



**Figure 16-38. Four-Time-Slice LCD Drive Waveform Examples Between SEG12 and Each Common Signals (1/3 Bias Method) (2/2)**

**(b) Waveform B**



**16.10.5 Eight-time-slice display example**

Figure 16-40 shows how the 15x8 dot LCD panel having the display pattern shown in Figure 16-39 is connected to the segment signals (SEG4 to SEG18) and the common signals (COM0 to COM7). This example displays data “123” in the LCD panel. The contents of the display data register (addresses F0404H to F0412H) correspond to this display.

The following description focuses on numeral “3.” ( 3 ) displayed in the first digit. To display “3.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG4 to SEG8 pins according to Table 16-19 at the timing of the common signals COM0 to COM7; see Figure 16-39 for the relationship between the segment signals and LCD segments.

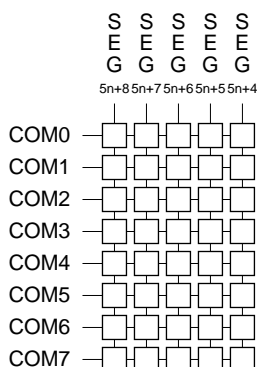
**Table 16-19. Select and Deselect Voltages (COM0 to COM7)**

Segment Common	SEG4	SEG5	SEG6	SEG7	SEG8
COM0	Select	Select	Select	Select	Select
COM1	Deselect	Select	Deselect	Deselect	Deselect
COM2	Deselect	Deselect	Select	Deselect	Deselect
COM3	Deselect	Select	Deselect	Deselect	Deselect
COM4	Select	Deselect	Deselect	Deselect	Deselect
COM5	Select	Deselect	Deselect	Deselect	Select
COM6	Deselect	Select	Select	Select	Deselect
COM7	Deselect	Deselect	Deselect	Deselect	Deselect

According to the above, it is determined that the display data register location (F0404H) that corresponds to SEG4 must contain 00110001.

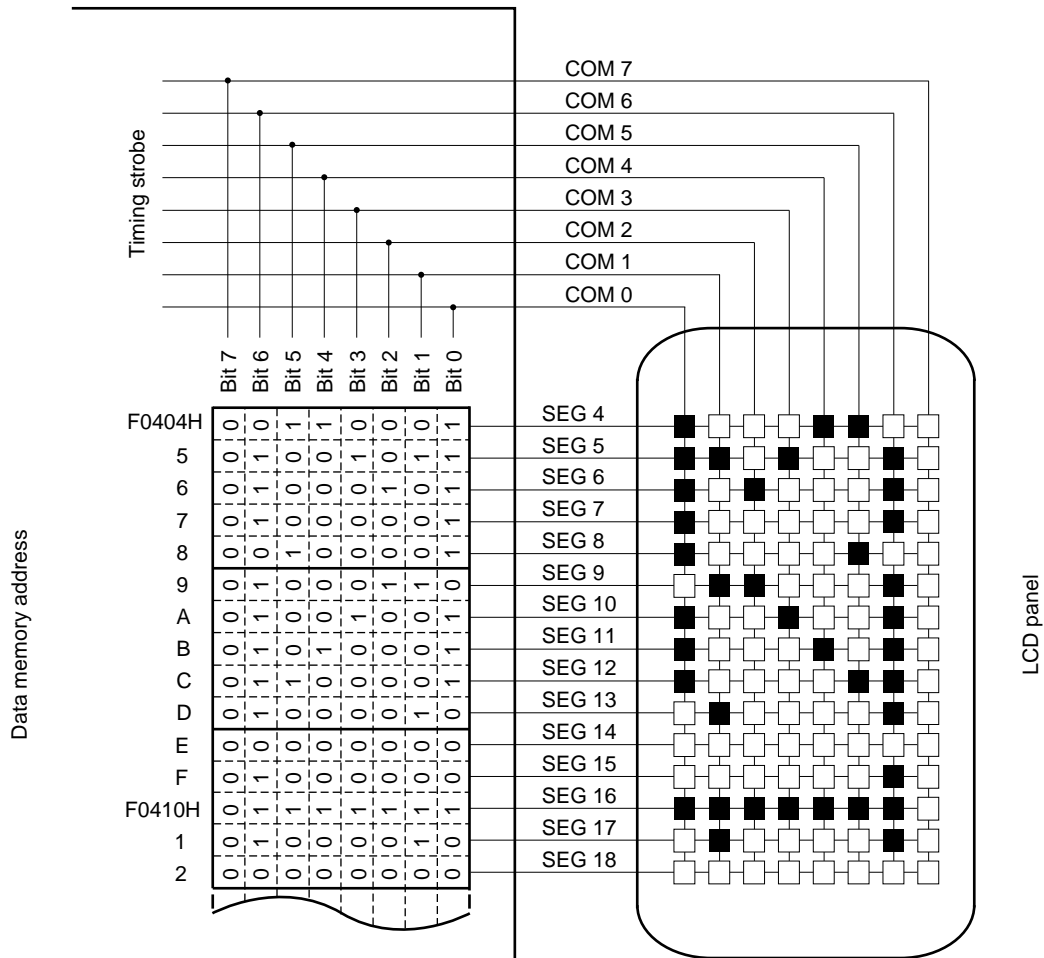
Figure 16-41 shows examples of LCD drive waveforms between the SEG4 signal and each common signal. When the select voltage is applied to SEG4 at the timing of COM0, a waveform is generated to turn on the corresponding LCD segment.

**Figure 16-39. Eight-Time-Slice LCD Display Pattern and Electrode Connections**



**Remark** 80-pin products: n = 0 to 8

Figure 16-40. Example of Connecting Eight-Time-Slice LCD Panel



**Figure 16-41. Eight-Time-Slice LCD Drive Waveform Examples Between SEG4 and Each Common Signals (1/4 Bias Method) (1/2)**

**(a) Waveform A**

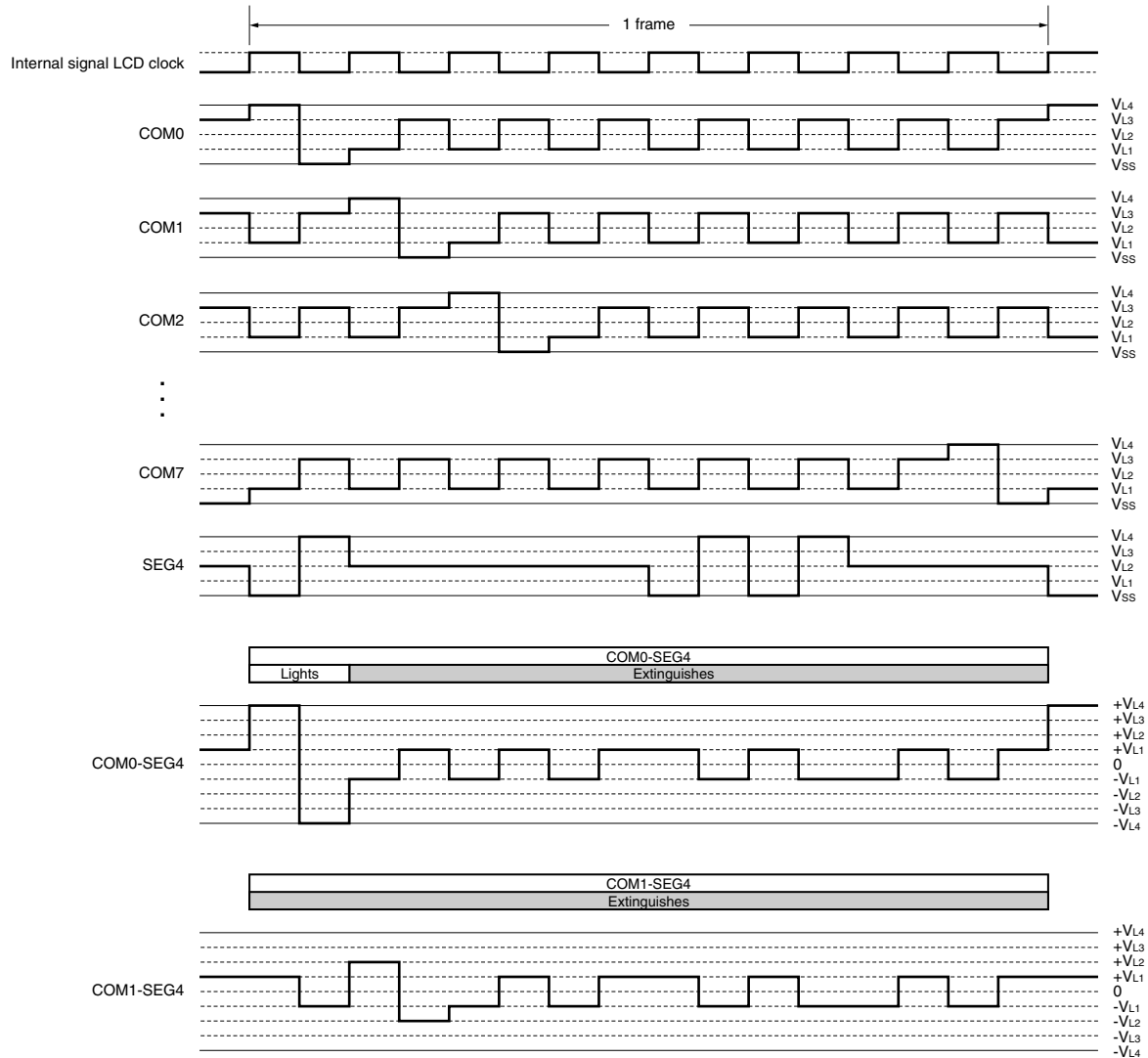
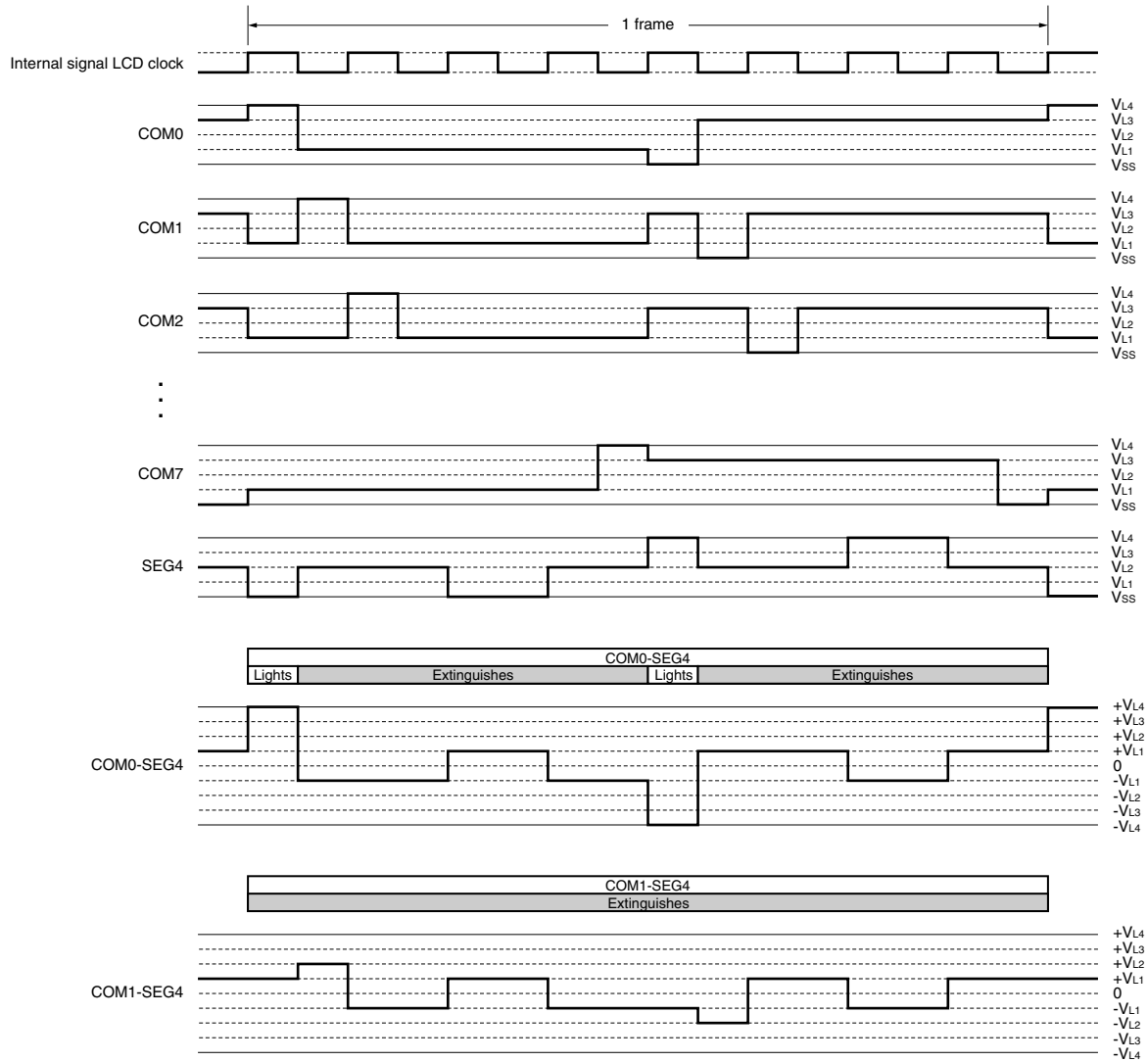


Figure 16-41. Eight-Time-Slice LCD Drive Waveform Examples Between SEG4 and Each Common Signals (1/4 Bias Method) (2/2)

(b) Waveform B



## CHAPTER 17 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR

### 17.1 Functions of Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator has the following functions.

- $16 \text{ bits} \times 16 \text{ bits} = 32 \text{ bits}$  (Unsigned)
- $16 \text{ bits} \times 16 \text{ bits} = 32 \text{ bits}$  (Signed)
- $16 \text{ bits} \times 16 \text{ bits} + 32 \text{ bits} = 32 \text{ bits}$  (Unsigned)
- $16 \text{ bits} \times 16 \text{ bits} + 32 \text{ bits} = 32 \text{ bits}$  (Signed)
- $32 \text{ bits} \div 32 \text{ bits} = 32 \text{ bits}$ , 32-bits remainder (Unsigned)

### 17.2 Configuration of Multiplier and Divider/Multiply-Accumulator

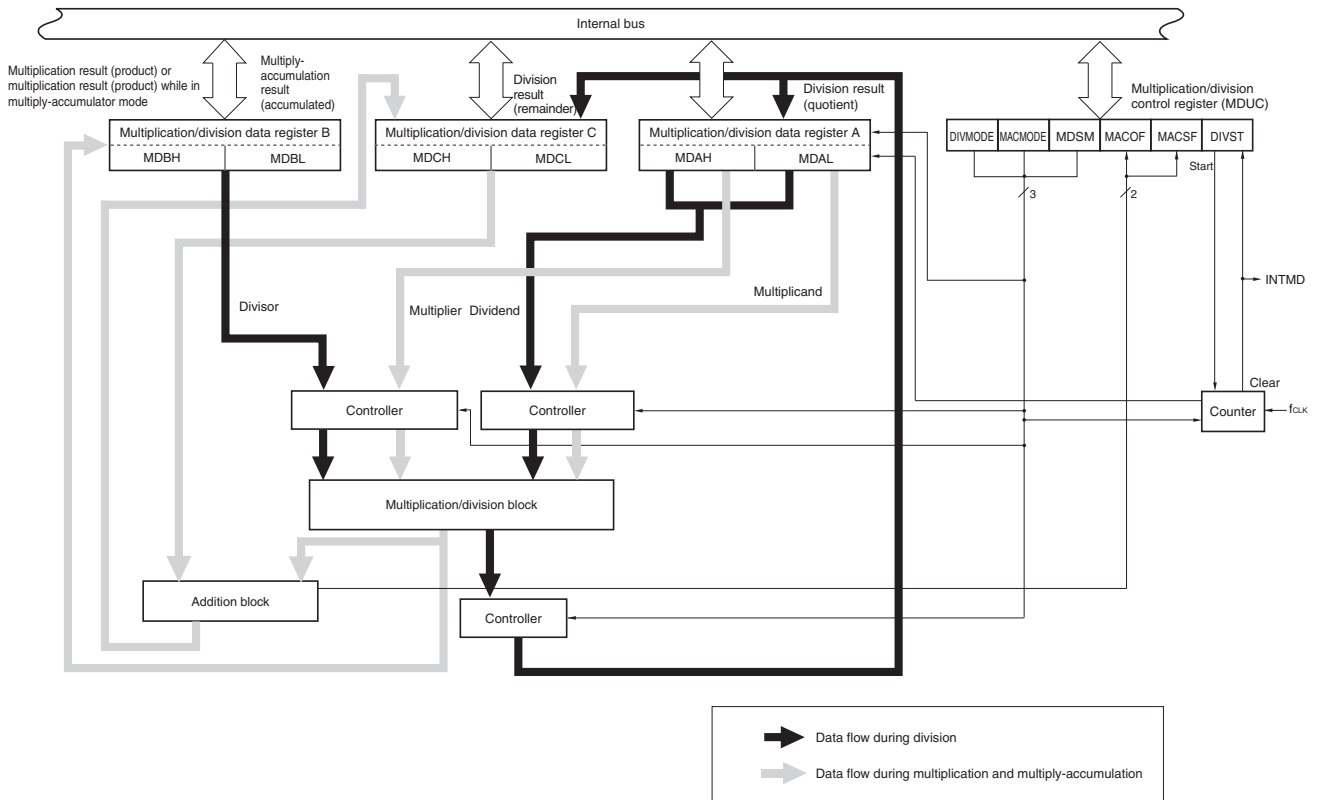
The multiplier and divider/multiply-accumulator consists of the following hardware.

**Table 17-1. 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 17-1 shows a block diagram of the multiplier and divider/multiply-accumulator.

Figure 17-1. Block Diagram of Multiplier and Divider/Multiply-Accumulator



**Remark** fclk: CPU/peripheral hardware clock frequency



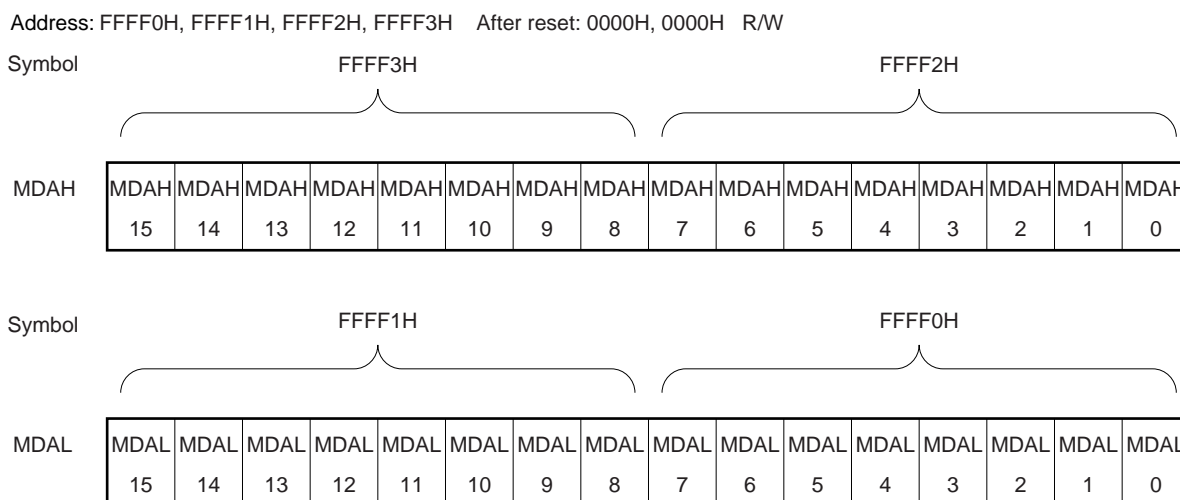
**17.2.1 Multiplication/division data register A (MDAH, MDAL)**

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 17-2. Format of Multiplication/Division Data Register A (MDAH, MDAL)**



- 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 17-2. 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

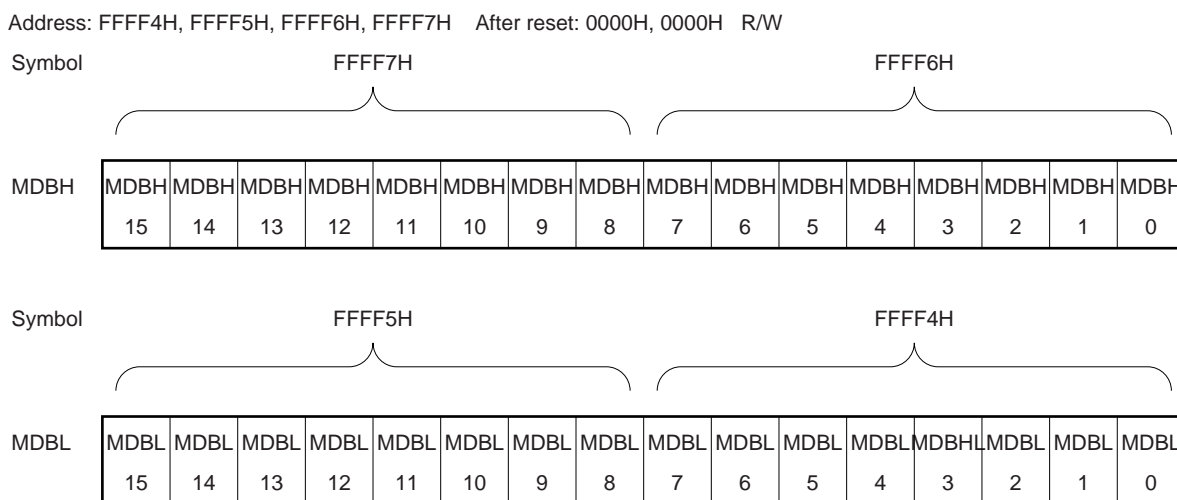
**17.2.2 Multiplication/division data register B (MDBL, MDBH)**

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 17-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)**



- 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 17-3. 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)	MDBH: Divisor (unsigned) (higher 16 bits) MDBL: Divisor (unsigned) (lower 16 bits)	–

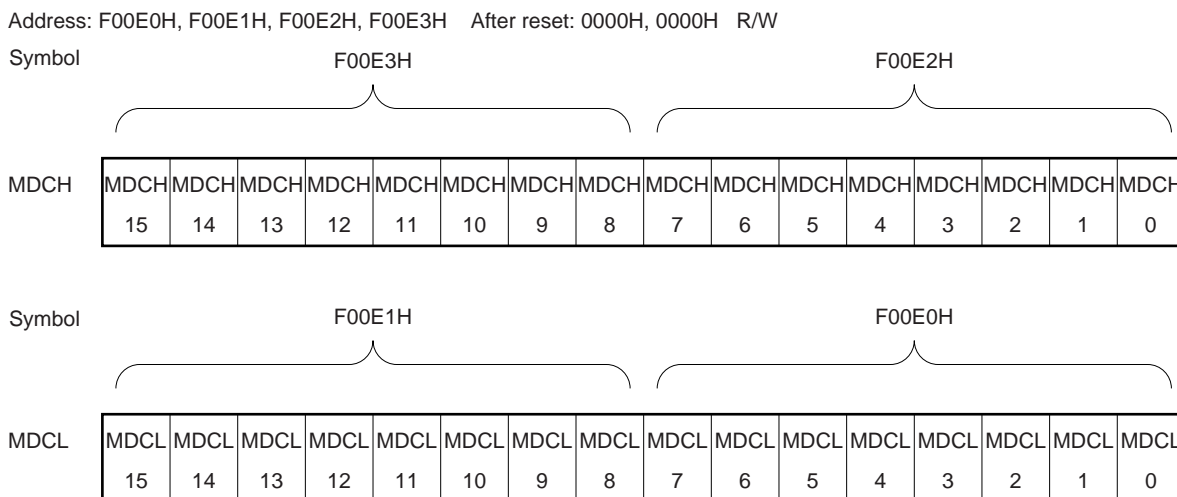
**17.2.3 Multiplication/division data register C (MDCL, MDCH)**

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 17-4. Format of Multiplication/Division Data Register C (MDCH, MDCL)**



- 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 17-4. 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

$$\begin{array}{ccc} \text{<Multiplier A>} & \text{<Multiplier B>} & \text{<Product>} \\ \text{MDAL (bits 15 to 0)} \times \text{MDAH (bits 15 to 0)} & = & [\text{MDBH (bits 15 to 0), MDBL (bits 15 to 0)}] \end{array}$$

- Register configuration during multiply-accumulation

$$\begin{array}{cccc} \text{<Multiplier A>} & \text{<Multiplier B>} & \text{< accumulated value >} & \text{< accumulated result >} \\ \text{MDAL (bits 15 to 0)} \times \text{MDAH (bits 15 to 0)} + \text{MDC (bits 31 to 0)} & = & [\text{MDCH (bits 15 to 0), MDCL (bits 15 to 0)}] \end{array}$$

(The multiplication result is stored in the MDBH (bits 15 to 0) and MDBL (bits 15 to 0).)

- Register configuration during division

$$\begin{array}{ccc} \text{<Dividend>} & & \text{<Divisor>} \\ [\text{MDAH (bits 15 to 0), MDAL (bits 15 to 0)}] \div [\text{MDBH (bits 15 to 0), MDBL (bits 15 to 0)}] & = & \\ \text{<Quotient>} & & \text{<Remainder>} \\ [\text{MDAH (bits 15 to 0), MDAL (bits 15 to 0)}] \dots [\text{MDCH (bits 15 to 0), MDCL (bits 15 to 0)}] \end{array}$$

### 17.3 Register Controlling Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator is controlled by using the multiplication/division control register (MDUC).

#### 17.3.1 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 17-5. Format of Multiplication/Division Control Register (MDUC)**

Address: F00E8H After reset: 00H R/W<sup>Note 1</sup>

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> <ul style="list-style-type: none"> <li>For the multiply-accumulator mode (unsigned) The bit is set when the accumulated value goes outside the range from 00000000h to FFFFFFFFh.</li> <li>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.</li> </ul>	

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 <sup>Note 2</sup>	Division operation start/stop
0	Division operation processing complete
1	Starts division operation/division operation processing in progress

(Note and Cautions are listed on the next page.)

**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).

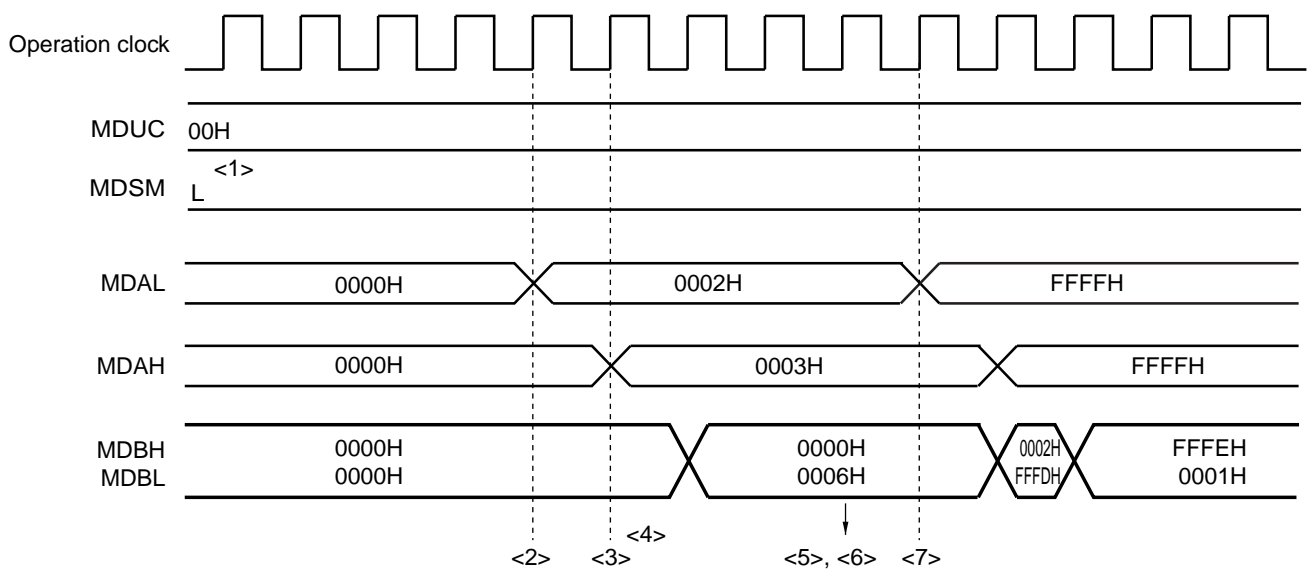
### 17.4 Operations of Multiplier and Divider/Multiply-Accumulator

#### 17.4.1 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 17-6.

**Figure 17-6. Timing Diagram of Multiplication (Unsigned) Operation (2 × 3 = 6)**



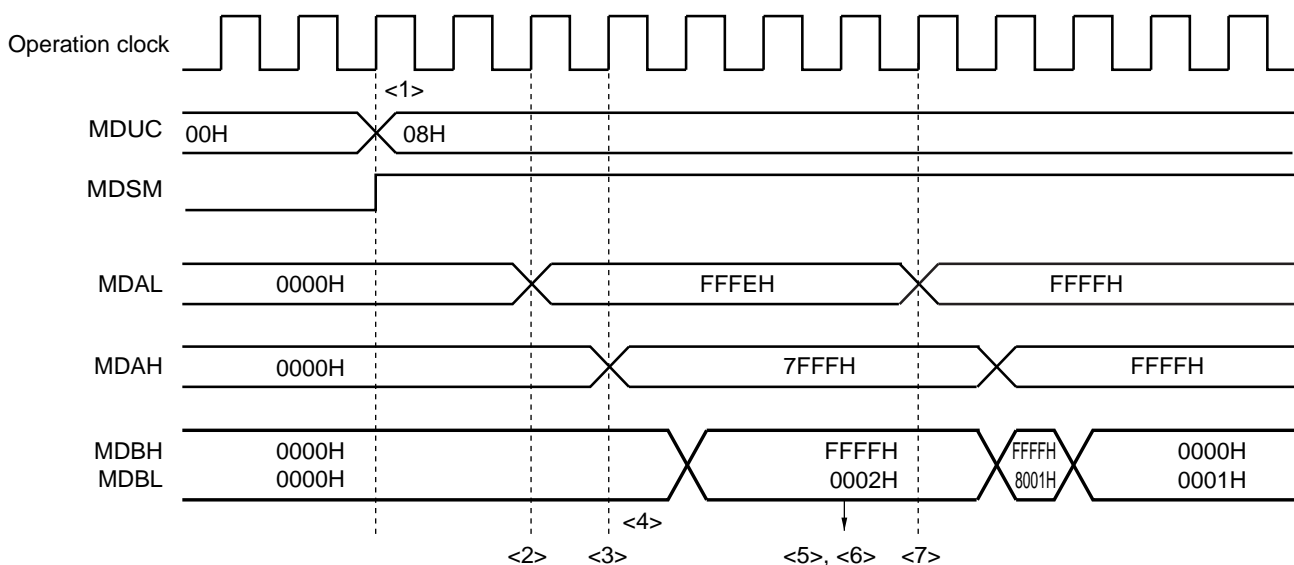
17.4.2 Multiplication (signed) operation

- 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 17-7.

Figure 17-7. Timing Diagram of Multiplication (Signed) Operation ( $-2 \times 32767 = -65534$ )



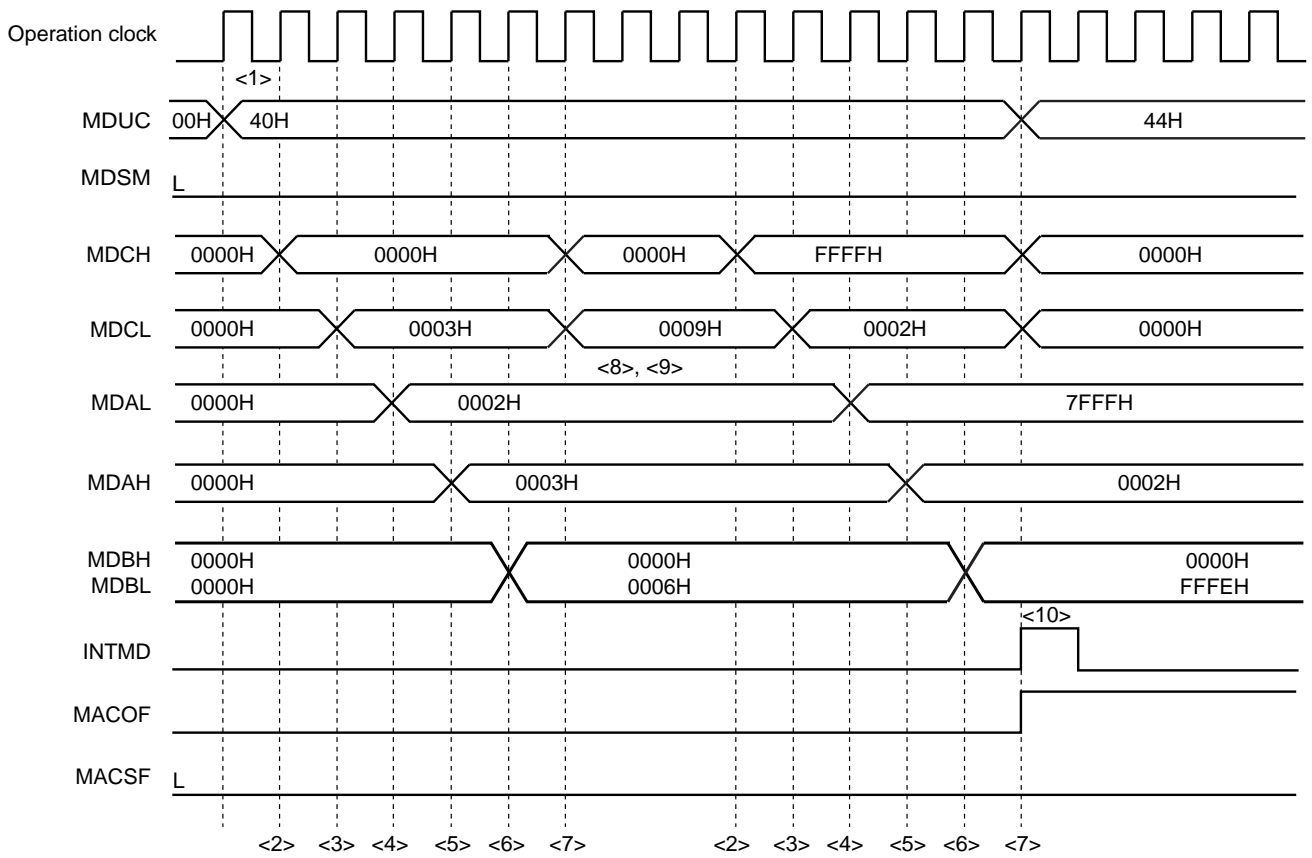


### 17.4.3 Multiply-accumulation (unsigned) operation

- 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, 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 17-8.

**Figure 17-8. Timing Diagram of Multiply-Accumulation (Unsigned) Operation**  
 $(2 \times 3 + 3 = 9 \rightarrow 32767 \times 2 + 4294901762 = 0 \text{ (over flow generated)})$



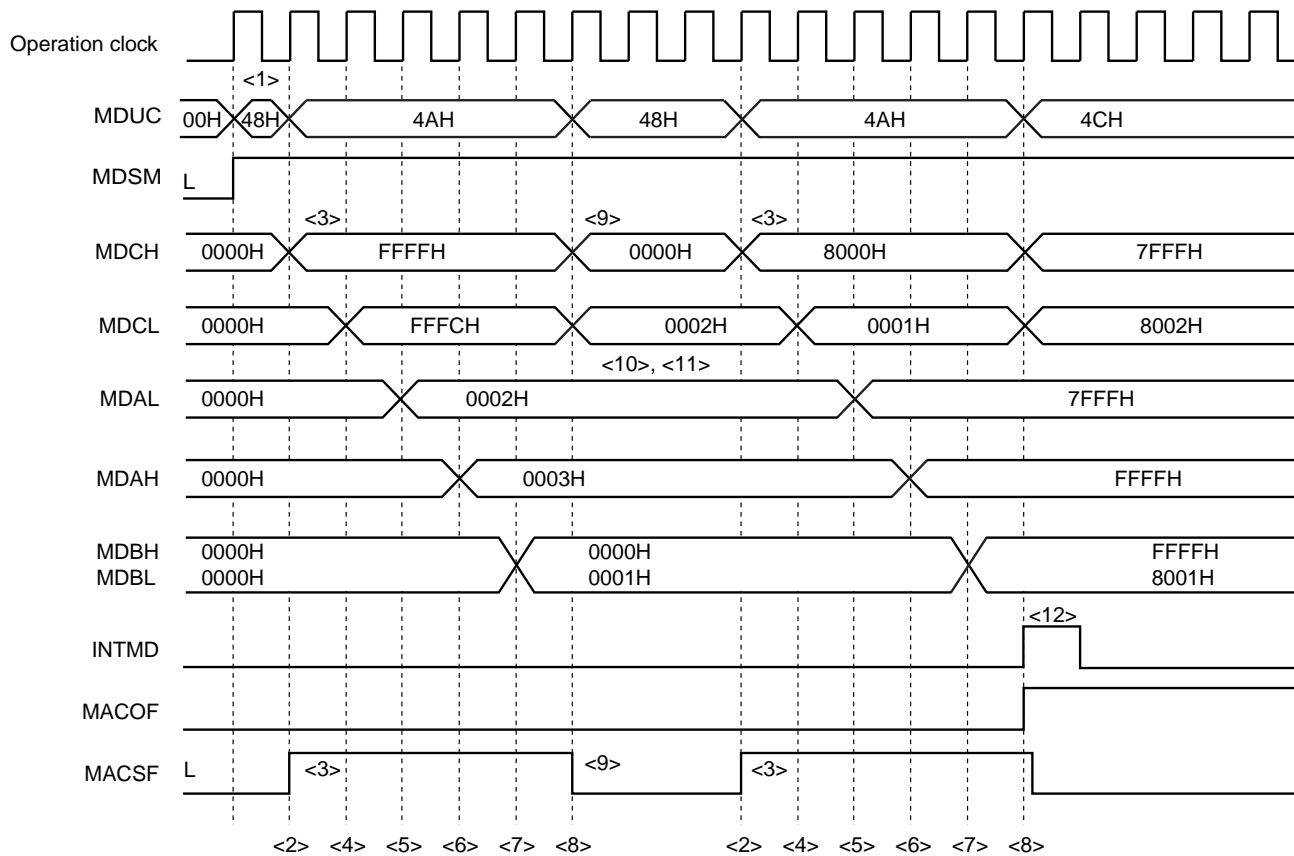
#### 17.4.4 Multiply-accumulation (signed) operation

- 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 17-9.

**Figure 17-9. Timing Diagram of Multiply-Accumulation (Signed) Operation**  
 $(2 \times 3 + (-4) = 2 \rightarrow 32767 \times (-1) + (-2147483647) = -2147516414$  (overflow occurs.))

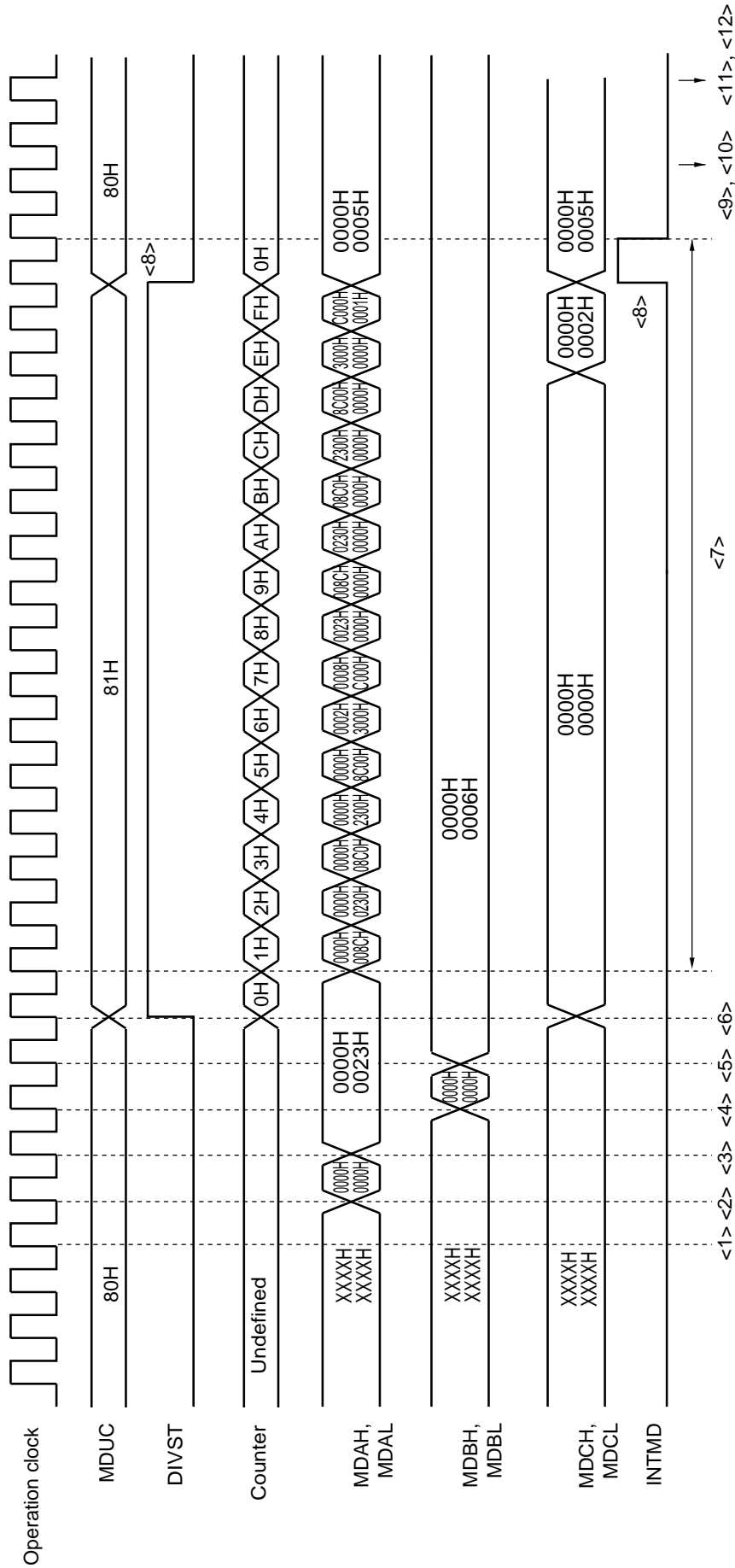


### 17.4.5 Division operation

- 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 17-10.

Figure 17-10. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



## CHAPTER 18 DMA CONTROLLER

The RL78/L13 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, and A/D can also be realized.

### 18.1 Functions of DMA Controller

- Number of DMA channels: 4 channels
- Transfer unit: 8 or 16 bits
- Maximum transfer unit: 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, CSI10, UART0 to UART3)
  - Timer array unit (channel 0, 1, 2, or 3)
  - 16-bit timer KB20
- 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

### 18.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

**Table 18-1. Configuration of DMA Controller**

Item	Configuration
Address registers	<ul style="list-style-type: none"> <li>• DMA SFR address registers 0 to 3 (DSA0 to DSA3)</li> <li>• DMA RAM address registers 0 to 3 (DRA0 to DRA3)</li> </ul>
Count register	<ul style="list-style-type: none"> <li>• DMA byte count registers 0 to 3 (DBC0 to DBC3)</li> </ul>
Control registers	<ul style="list-style-type: none"> <li>• DMA mode control registers 0 to 3 (DMC0 to DMC3)</li> <li>• DMA operation control registers 0 to 3 (DRC0 to DRC3)</li> </ul>

#### 18.2.1 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.

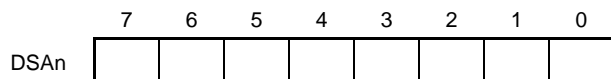
In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

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 18-1. Format of DMA SFR Address Register n (DSAn)**

Address: FFFB0H (DSA0), FFFB1H (DSA1), F0200H (DSA2), F0201H (DSA3) After reset: 00H R/W



**Remark** n: DMA channel number (n = 0 to 3)



### 18.2.2 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 (see **Table 18-2**) can be set to this register.

Set the lower 16 bits of the RAM address.

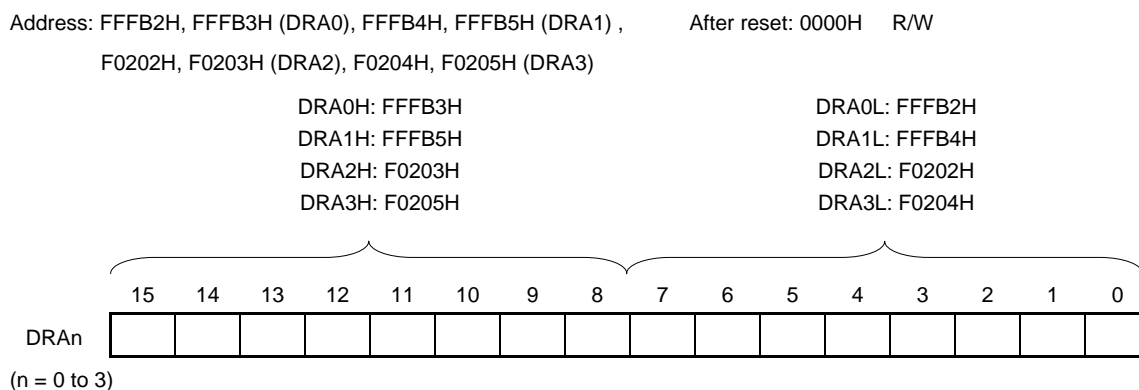
This register is automatically incremented when DMA transfer has been started. It is incremented by +1 in the 8-bit transfer mode and by +2 in the 16-bit transfer mode. DMA transfer is started from the address set to this DRAn register. When the data of the last address has been transferred, the DRAn 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 18-2. Format of DMA RAM Address Register n (DRAn)**



**Table 18-2. Internal RAM Area other than the General-purpose Registers**

Part Number	Internal RAM Area Other than the General-purpose Registers
R5F10WLA, R5F10WMA	FFB00H to FFEDFH
R5F10WLC, R5F10WMC	FF900H to FFEDFH
R5F10WLD, R5F10WMD	FF700H to FFEDFH
R5F10WLE, R5F10WME	FEF00H to FFEDFH
R5F10WLF, R5F10WMF	FF700H to FFEDFH
R5F10WLG, R5F10WMG	FD00H to FFEDFH

**Remark** n: DMA channel number (n = 0 to 3)

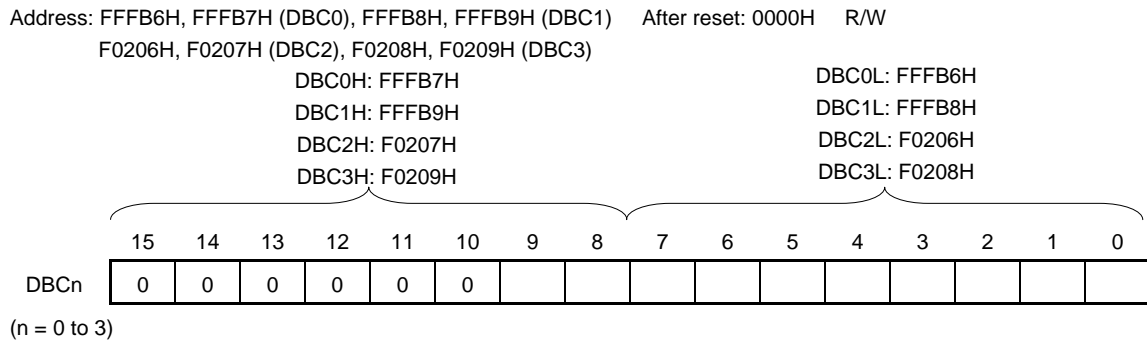
**18.2.3 DMA byte count register n (DBCn)**

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 18-3. Format of DMA Byte Count Register n (DBCn)**



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 to 3)

### 18.3 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 to 3)

### 18.3.1 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 18-4. Format of DMA Mode Control Register n (DMCn) (1/3)**

Address: FFFBAH (DMC0), FFFBBH (DMC1), F020AH (DMC2), F020BH (DMC3) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0
STGn <sup>Note 1</sup>	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 <sup>Note 2</sup>	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**
- The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.
  - When DMA transfer is held pending while using two or more DMA channels, be sure to hold the DMA transfer pending for all channels (by setting the DWAIT0, DWAIT1, DWAIT2, and DWAIT3 bits to 1).

**Remark** n: DMA channel number (n = 0 to 3)

**Figure 18-4. Format of DMA Mode Control Register n (DMCn) (2/3)**

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)

				Selection of DMA start source <sup>Note</sup>	
IFCn 3	IFCn 2	IFCn 1	IFCn 0	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 00 count or capture end interrupt
0	0	1	1	INTTM01	End of timer channel 01 count or capture end interrupt
0	1	0	0	INTTM02	End of timer channel 02 count or capture end interrupt
0	1	0	1	INTTM03	End of timer channel 03 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	UART0 reception transfer end interrupt
1	0	0	0	INTST1/INTCSI10	UART1 transmission transfer end or buffer empty interrupt/CSI10 transfer end or buffer empty interrupt
1	0	0	1	INTSR1	UART1 reception transfer end interrupt
1	0	1	0	INTST2	UART2 transmission transfer end or buffer empty interrupt
1	0	1	1	INTSR2	UART2 reception transfer end interrupt
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)

Figure 18-4. Format of DMA Mode Control Register n (DMCn) (3/3)

Address: F020AH (DMC2), F020BH (DMC3) 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 = 2 or 3)

IFCn 3	IFCn 2	IFCn 1	IFCn 0	Selection of DMA start source <sup>Note 1</sup>	
				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	INTTKB2	Timer KB20 count end interrupt
0	1	1	0	INTST3 <sup>Note 2</sup>	UART3 transmission transfer end or buffer empty interrupt
0	1	1	1	INTSR3 <sup>Note 2</sup>	UART3 reception transfer end interrupt
1	0	0	0	INTST1/INTCSI10	UART1 transmission transfer end or buffer empty interrupt/CSI10 transfer end or buffer empty interrupt
1	0	0	1	INTSR1	UART1 reception transfer end interrupt
1	0	1	0	INTST2	UART2 transmission transfer end or buffer empty interrupt
1	0	1	1	INTSR2	UART2 reception transfer end interrupt
Other than above				Setting prohibited	

- Notes**
1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.
  2. 80-pin products only.

**Remark** n: DMA channel number (n = 2, 3)

**18.3.2 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 18-5. Format of DMA Operation Control Register n (DRCn)**

Address: FFFBCH (DRC0), FFFBDH (DRC1), F020CH (DRC2), F020DH (DRC3) 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, see 18.5.5 Forced termination by software).

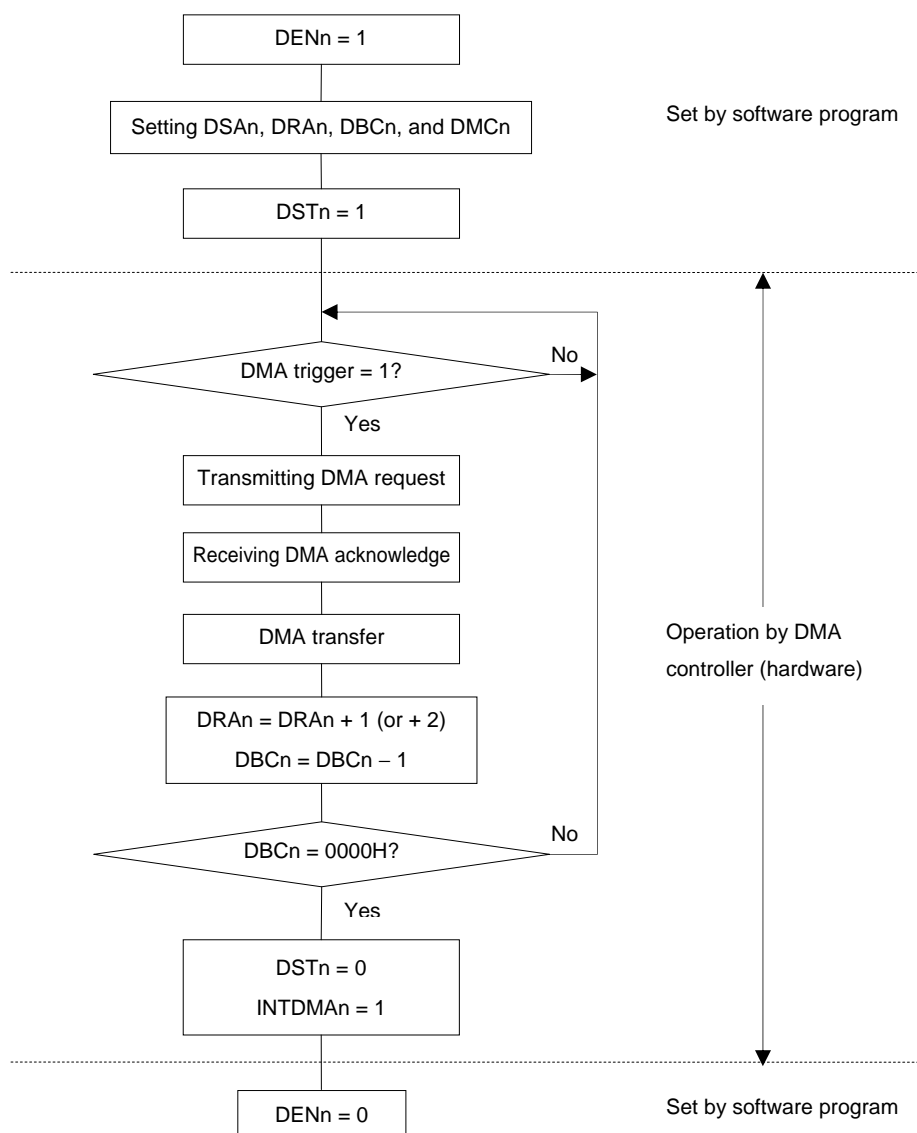
**Remark** n: DMA channel number (n = 0 to 3)

### 18.4 Operation of DMA Controller

#### 18.4.1 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 (INTDMA n).
- <6> Stop the operation of the DMA controller by clearing the DENn bit to 0 when the DMA controller is not used.

**Figure 18-6. Operation Procedure**



**Remark** n: DMA channel number (n = 0 to 3)



### 18.4.2 Transfer mode

The following four modes can be selected for DMA transfer by using bits 6 and 5 (DRSn and DS<sub>n</sub>) of DMA mode control register n (DMCn).

DRS <sub>n</sub>	DS <sub>n</sub>	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.

### 18.4.3 Termination of DMA transfer

When DBC<sub>n</sub> = 00H and DMA transfer is completed, the DST<sub>n</sub> bit is automatically cleared to 0. An interrupt request (INTDMA<sub>n</sub>) is generated and transfer is terminated.

When the DST<sub>n</sub> bit is cleared to 0 to forcibly terminate DMA transfer, DMA byte count register n (DBC<sub>n</sub>) and DMA RAM address register n (DRAn) hold the value when transfer is terminated.

The interrupt request (INTDMA<sub>n</sub>) is not generated if transfer is forcibly terminated.

**Remark** n: DMA channel number (n = 0 to 3)

## 18.5 Example of Setting of DMA Controller

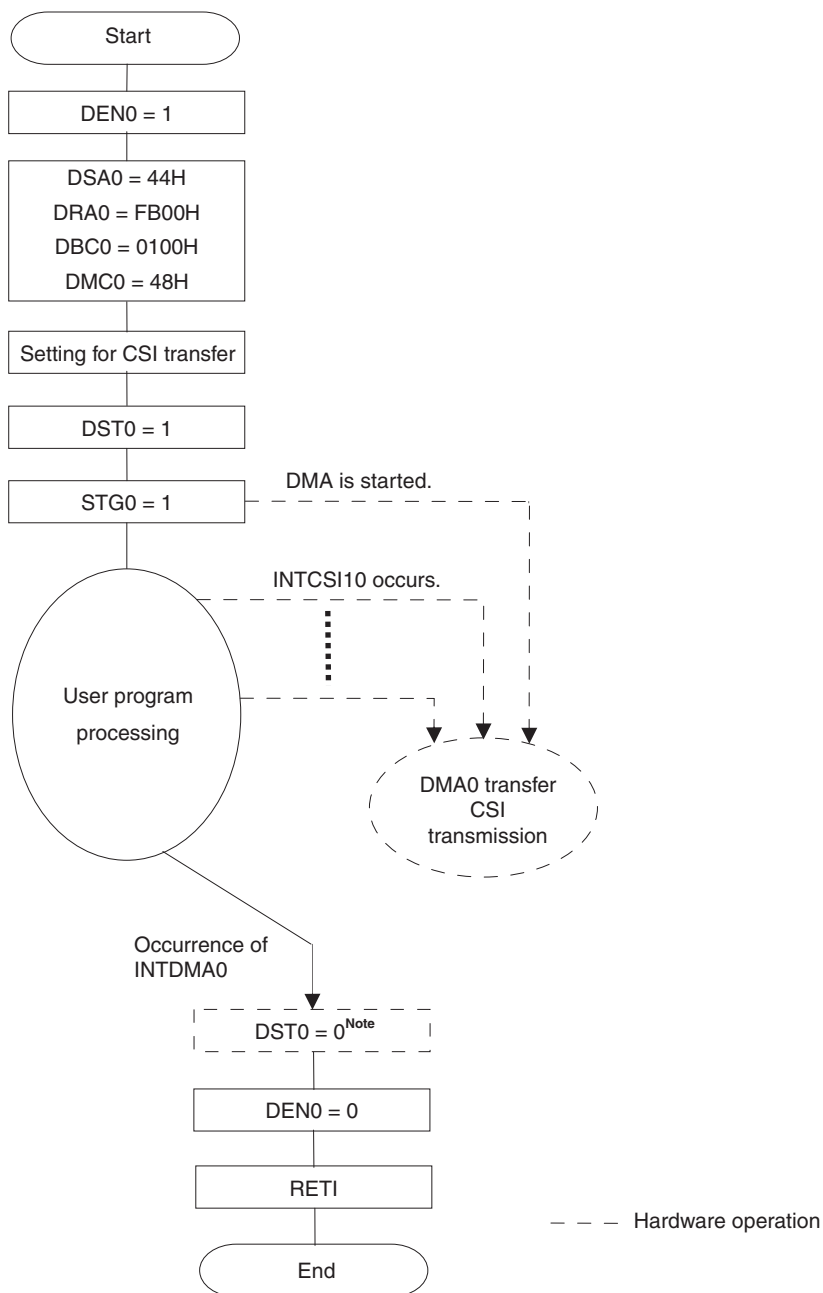
### 18.5.1 CSI consecutive transmission

A flowchart showing an example of setting for CSI consecutive transmission is shown below.

- Consecutive transmission of CSI10 (256 bytes)
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI10 (software trigger (STG0) only for the first start source)
- Interrupt of CSI10 is specified by IFC03 to IFC00 = 1000B.
- Transfers FFB00H to FFBFFH (256 bytes) of RAM to FFF44H of the data register (SIO10) of CSI.

**Remark** IFC03 to IFC00: Bits 3 to 0 of DMA mode control registers 0 (DMC0)

Figure 18-7. Example of Setting for 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, see 18.5.5 Forced termination by software).

The first trigger for consecutive transmission is not started by the interrupt of CSI. In this example, it starts by a software trigger.

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.

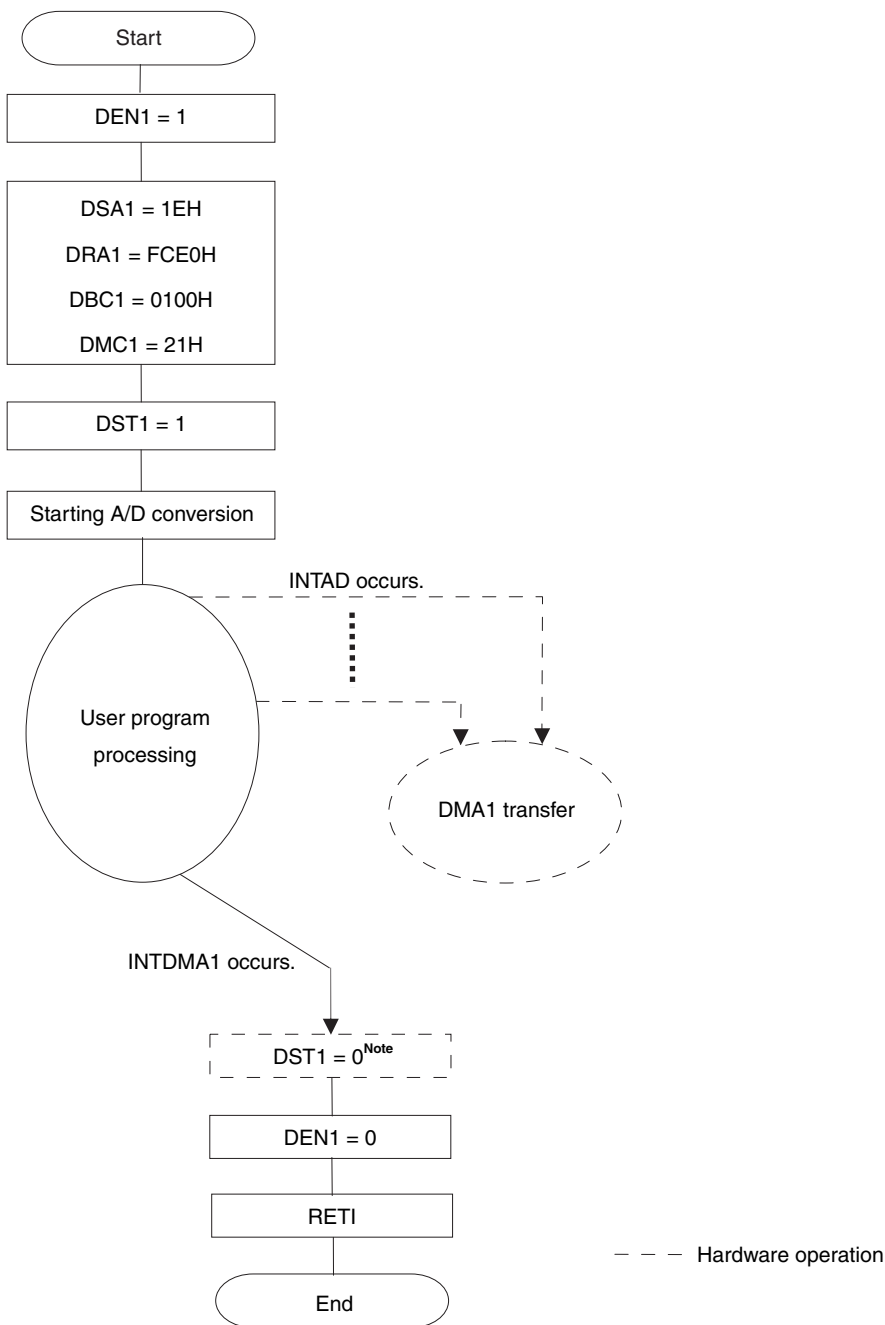
### 18.5.2 Consecutive capturing of A/D conversion results

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 18-8. Example of Setting of Consecutively Capturing A/D Conversion Results



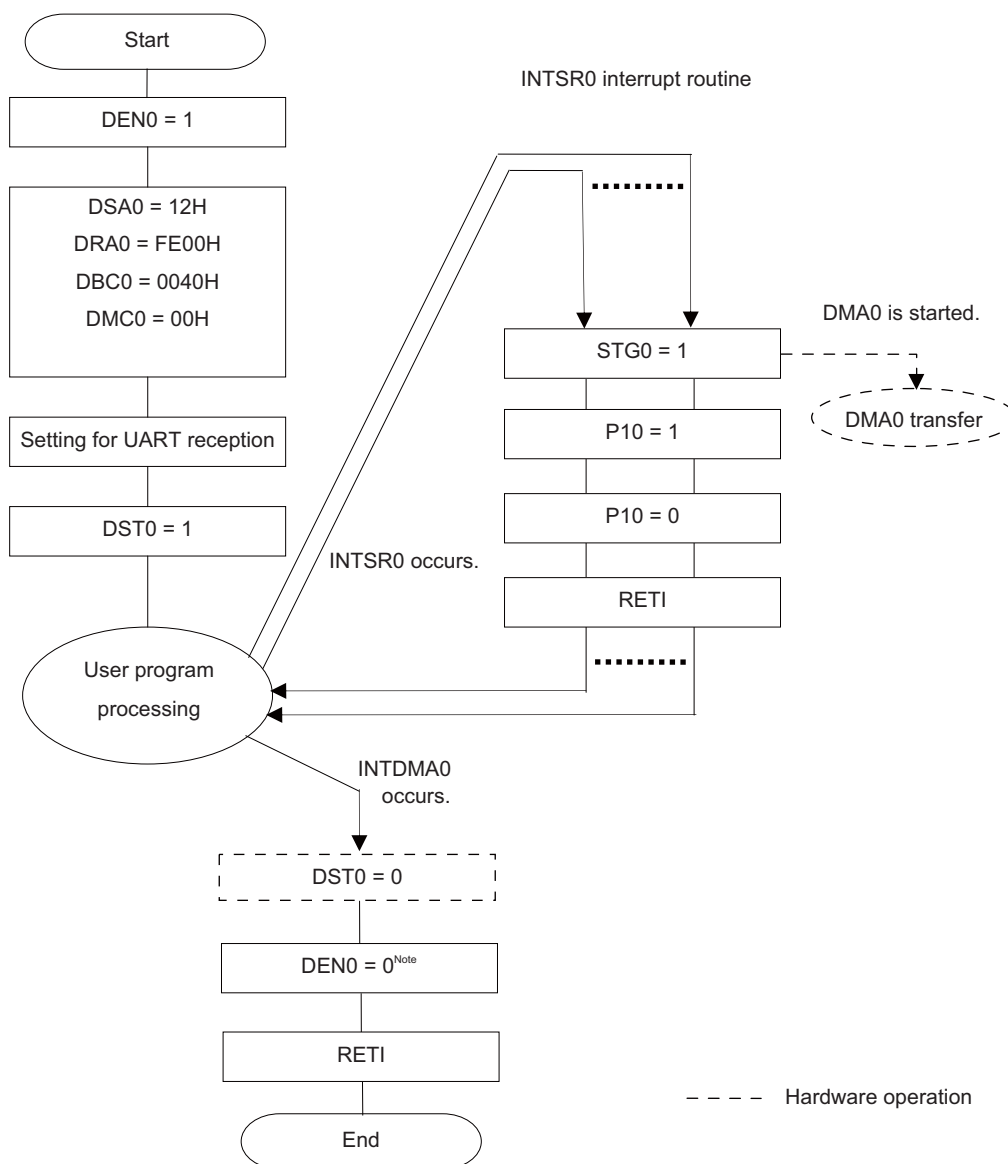
**Note** The DST1 flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set the DST1 bit to 0 and then the DEN1 bit to 0 (for details, see 18.5.5 Forced termination by software).

**18.5.3 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 P10 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 18-9. 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, see 18.5.5 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.

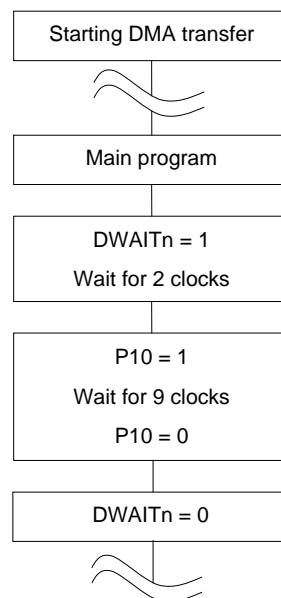
#### 18.5.4 Holding DMA transfer pending by DWAITn bit

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 P10 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 18-10. Example of Setting for Holding DMA Transfer Pending by DWAITn Bit**



**Caution** When DMA transfer is held pending while using two or more DMA channels, be sure to held the DMA transfer pending for all channels (by setting DWAIT0, DWAIT1, DWAIT2, and DWAIT3 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 to 3)
  2. 1 clock:  $1/f_{CLK}$  ( $f_{CLK}$ : CPU clock)

**18.5.5 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 (INTDMA<sub>n</sub>) of DMA<sub>n</sub>, therefore, perform either of the following processes.

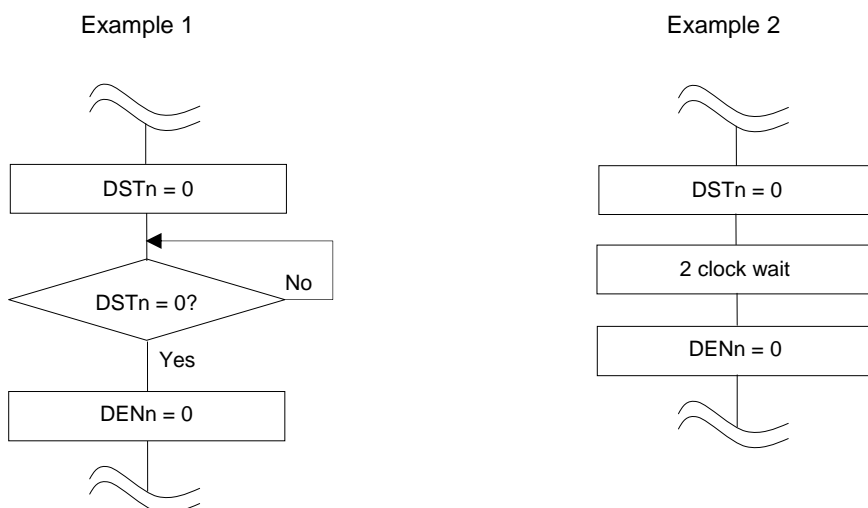
<When using one DMA channel>

- 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 or more DMA channels>

- To forcibly terminate DMA transfer by software when using two or more DMA channels (by setting DSTn to 0), clear the DSTn bit to 0 after the DMA transfer is held pending by setting the DWAITn bits of all using channels to 1. Next, clear the DWAITn bits of all using channels to 0 to cancel the pending status, and then clear the DENn bit to 0.

**Figure 18-11. Forced Termination of DMA Transfer (1/2)**



- Remarks**
1. n: DMA channel number (n = 0 to 3)
  2. 1 clock: 1/f<sub>CLK</sub> (f<sub>CLK</sub>: CPU clock)

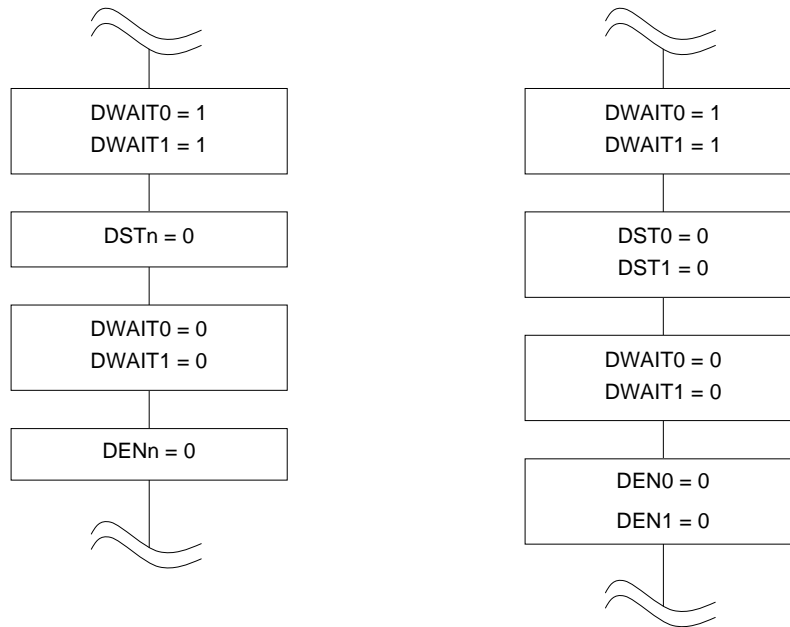


Figure 18-11. Forced Termination of DMA Transfer (2/2)

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<sub>CLK</sub> (f<sub>CLK</sub>: CPU clock)

## 18.6 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 > DMA channel 2 > DMA channel 3. 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 18-3. Response Time of DMA Transfer**

	Minimum Time	Maximum Time
Response time	3 clocks	10 clocks <sup>Note</sup>

**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 18.6 (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 18-4. 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.

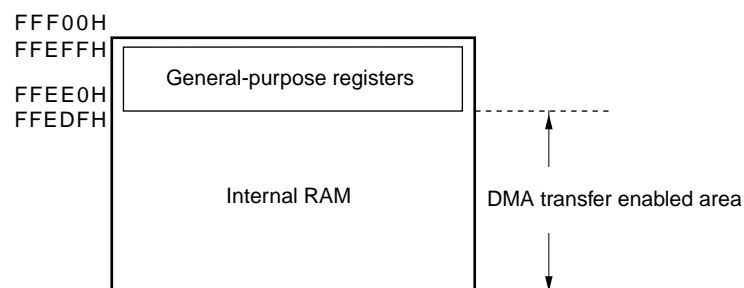
- CALL !addr16
- CALL \$!addr20
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- POP PSW
- BTCLR PSW. bit, \$addr20
- EI
- DI
- Write instructions for registers IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L 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) Operation if instructions for accessing the data flash area**

- If the data flash area is accessed one instruction execution after 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 19 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.

		64-pin	80-pin
Maskable interrupts	External	11	
	Internal	32	35

### 19.1 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, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L).

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 19-1**.

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.

### 19.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Table 19-1**). 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 19-1. Interrupt Source List (1/3)**

Interrupt Type	Default Priority <sup>Note 1</sup>	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type <sup>Note 2</sup>	80-pin	64-pin
		Name	Trigger					
<R> Maskable	0	INTWDTI	Watchdog timer interval <sup>Note 3</sup> (75% of overflow time+1/2f <sub>IL</sub> )	Internal	00004H	(A)	√	√
	1	INTLVI	Voltage detection <sup>Note 4</sup>		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	INTST2	UART2 transmission transfer end or buffer empty interrupt	Internal	00014H	(A)	√	√
	9	INTSR2/ INTCSI21/ INTIIC21	UART2 reception transfer end/CSI21 transfer end or buffer empty interrupt/IIC21 transfer end		00016H		√	√
	10	INTSRE2	UART2 reception communication error occurrence		00018H		√	√
	11	INTDMA0	End of DMA0 transfer		0001AH		√	√
	12	INTDMA1	End of DMA1 transfer		0001CH		√	√
	13	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end		0001EH		√	√
	14	INTTM00	End of timer channel 00 count or capture		00020H		√	√
	15	INTSR0	UART0 reception transfer end		00022H		√	√
	16	INTSRE0	UART0 reception communication error occurrence		00024H		√	√
		INTTM01H	End of timer channel 01 count or capture (at higher 8-bit timer operation)				√	√
	17	INTST1/ INTCSI10/ INTIIC10	UART1 transmission transfer end or buffer empty interrupt/CSI10 transfer end or buffer empty interrupt/IIC10 transfer end		00026H		√	√
	18	INTSR1	UART1 reception transfer end		00028H		√	√
	19	INTSRE1	UART1 reception communication error occurrence		0002AH		√	√
		INTTM03H	End of timer channel 03 count or capture (at higher 8-bit timer operation)				√	√
20	INTIICA0	End of IICA0 communication	0002CH	√	√			
21	INTRTIT	RTC2 correction timing	0002EH	√	√			
22	INTTM01	End of timer channel 01 count or capture (at 16-bit/lower 8-bit timer operation)	00032H	√	√			

- Notes**
1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 44 indicates the lowest priority.
  2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 19-1.
  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 19-1. Interrupt Source List (2/3)

Interrupt Type	Default Priority <sup>Note 1</sup>	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type <sup>Note 2</sup>	80-pin	64-pin
		Name	Trigger					
<R> Maskable	23	INTTM02	End of timer channel 02 count or capture	Internal	00034H	(A)	√	√
	24	INTTM03	End of timer channel 03 count or capture (at 16-bit/lower 8-bit timer operation)		00036H		√	√
	25	INTAD	End of A/D conversion		00038H		√	√
	26	INTRTC	Fixed-cycle signal of real-time clock 2/alarm match detection		0003AH		√	√
	27	INTIT	Interval signal of 12-bit interval timer detection		0003CH		√	√
	28	INTKR	Key return signal detection	External	0003EH	(C)	√	√
	29	INTST3	UART3 transmission transfer end or buffer empty interrupt	Internal	00040H	(A)	√	–
	30	INTSR3	UART3 reception transfer end		00042H		√	–
	31	INTTKB20	End of timer KB20 count		00044H		√	√
	32	INTTM04	End of timer channel 04 count or capture		00046H		√	√
	33	INTTM05	End of timer channel 05 count or capture		00048H		√	√
	34	INTP6	Pin input edge detection	External	0004AH	(B)	√	√
	35	INTP7			0004CH		√	√
	36	INTCMP0	Comparator detection 0		00050H		√	√
	37	INTCMP1	Comparator detection 1		00052H		√	√
	38	INTTM06	End of timer channel 06 count or capture	Internal	00054H	(A)	√	√
	39	INTTM07	End of timer channel 07 count or capture		00056H		√	√
	40	INTSRE3	UART3 reception communication error occurrence		0005CH		√	–
	41	INTMD	End of division operation/overflow occur		0005EH		√	√
	42	INTFL	Reserved <sup>Note 3</sup>		00062H		√	√
	43	INTDMA2	End of DMA2 transfer		00064H		√	√
	44	INTDMA3	End of DMA3 transfer		00066H		√	√

- Notes**
1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 44 indicates the lowest priority.
  2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 19-1.
  3. Be used at the flash self programming library or the data flash library.

Table 19-1. Interrupt Source List (3/3)

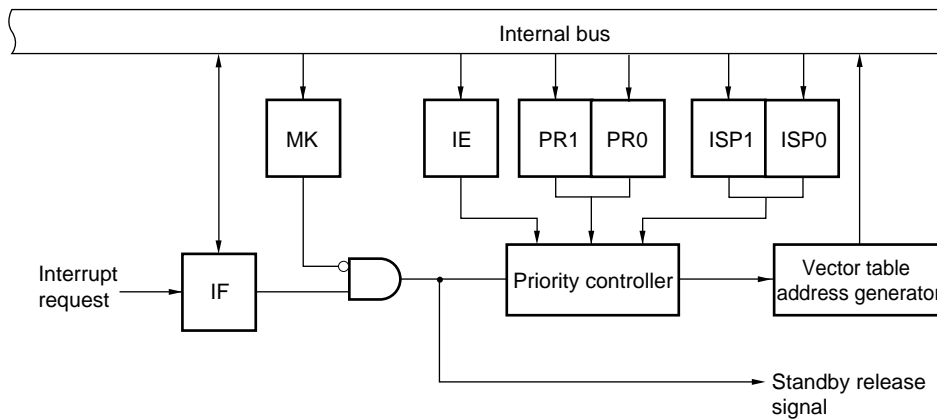
Interrupt Type	Default Priority <sup>Note 1</sup>	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type <sup>Note 2</sup>	80-pin	64-pin
Software	–	BRK	Execution of BRK instruction	–	0007EH	(D)	√	√
Reset	–	RESET	RESET pin input	–	00000H	–	√	√
		POR	Power-on-reset				√	√
		LVD	Voltage detection <sup>Note 3</sup>				√	√
		WDT	Overflow of watchdog timer				√	√
		TRAP	Execution of illegal instruction <sup>Note 4</sup>				√	√
		IAW	Illegal-memory access				√	√
		RPE	RAM parity error				√	√

- Notes**
- The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 44 indicates the lowest priority.
  - Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 19-1.
  - When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.
  - 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.

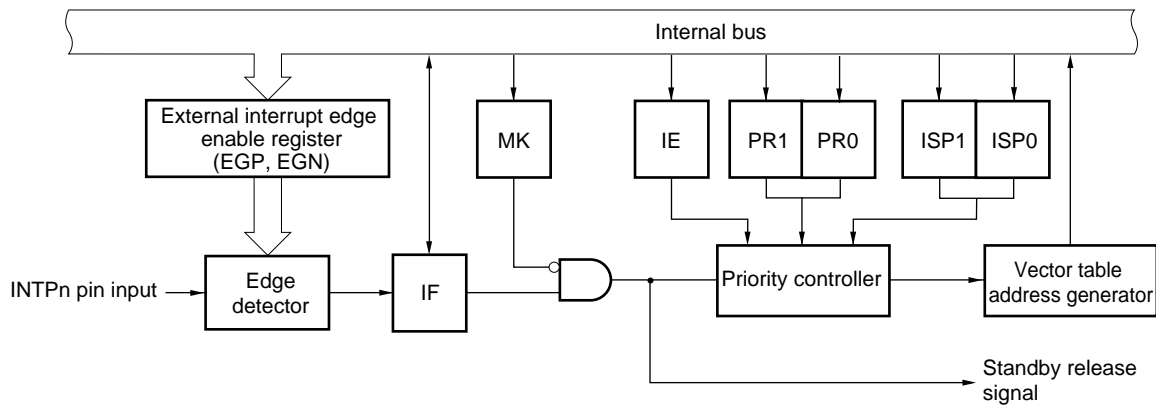


Figure 19-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)

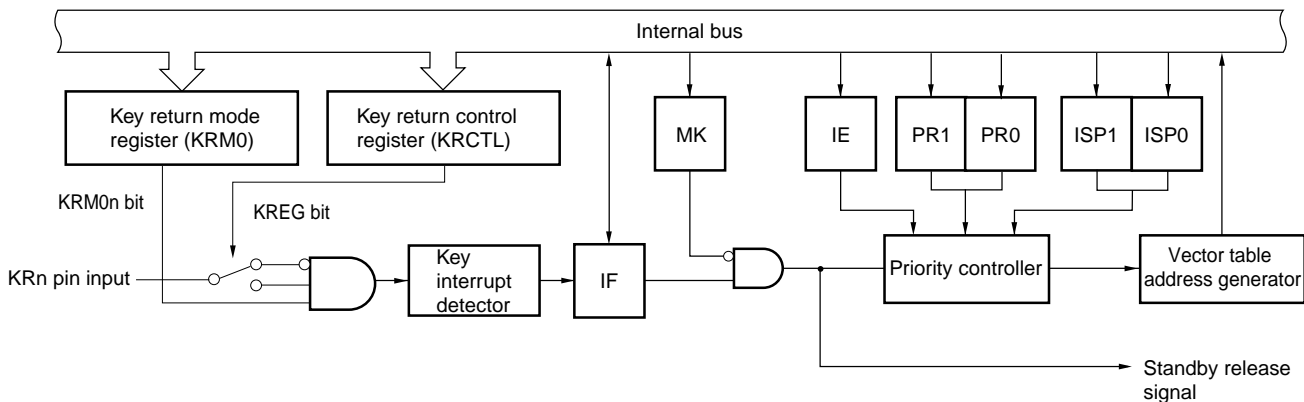


- IF: Interrupt request flag
- IE: Interrupt enable flag
- ISP0: In-service priority flag 0
- ISP1: In-service priority flag 1
- MK: Interrupt mask flag
- PR0: Priority specification flag 0
- PR1: Priority specification flag 1

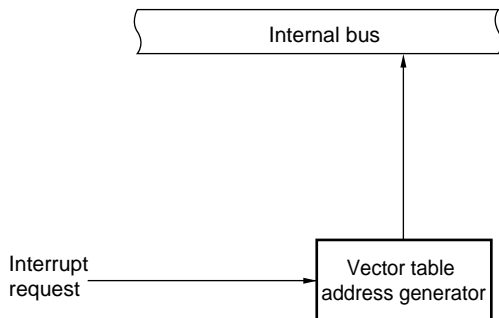
**Remark** n = 0 to 7

Figure 19-1. 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** 64-pin: n = 0 to 4, 7  
 80-pin: n = 0 to 7

### 19.3 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, IF3L)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L)
- External interrupt rising edge enable register (EGP0)
- External interrupt falling edge enable register (EGN0)
- Program status word (PSW)

Table 19-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

**Table 19-2. Flags Corresponding to Interrupt Request Sources (1/3)**

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		80-pin	64-pin
		Register		Register		Register		
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	√	√
INTLVI	LVIIIF		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		√	√
INTST2	STIF2	IF0H	STMK2	MK0H	STPR02, STPR12	PR00H, PR10H	√	√
INTSR2	SRIF2		SRMK2		SRPR02, SRPR12		√	√
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12		√	√
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10		√	√
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11		√	√
INTST0 <sup>Note</sup>	STIF0 <sup>Note</sup>		STMK0 <sup>Note</sup>		STPR00, STPR10 <sup>Note</sup>		√	√
INTCSI00 <sup>Note</sup>	CSIIIF00 <sup>Note</sup>		CSIMK00 <sup>Note</sup>		CSIPR000, CSIPR100 <sup>Note</sup>		√	√
INTIIC00 <sup>Note</sup>	IICIF00 <sup>Note</sup>		IICMK00 <sup>Note</sup>		IICPR000, IICPR100 <sup>Note</sup>		√	√
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		√	√
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10		√	√

**Note** 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.

Table 19-2. Flags Corresponding to Interrupt Request Sources (2/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		80-pin	64-pin			
		Register		Register		Register					
INTSRE0 <sup>Note 1</sup>	SREIF0 <sup>Note 1</sup>	IF1L	SREMK0 <sup>Note 1</sup>	MK1L	SREPR0, SREPR10 <sup>Note 1</sup>	PR01L, PR11L	√	√			
INTTM01H <sup>Note 1</sup>	TMIF01H <sup>Note 1</sup>		TMMK01H <sup>Note 1</sup>		TMPR001H, TMPR101H <sup>Note 1</sup>		√	√			
INTST1 <sup>Note 2</sup>	STIF1 <sup>Note 2</sup>		STMK1 <sup>Note 2</sup>		STPR01, STPR11 <sup>Note 2</sup>		√	√			
INTCSI10 <sup>Note 2</sup>	CSIF10 <sup>Note 2</sup>		CSIMK10 <sup>Note 2</sup>		CSIPR010, CSIPR110 <sup>Note 2</sup>		√	√			
INTIIC10 <sup>Note 2</sup>	IICIF10 <sup>Note 2</sup>		IICMK10 <sup>Note 2</sup>		IICPR010, IICPR110 <sup>Note 2</sup>		√	√			
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11		√	√			
INTSRE1 <sup>Note 3</sup>	SREIF1 <sup>Note 3</sup>		SREMK1 <sup>Note 3</sup>		SREPR01, SREPR11 <sup>Note 3</sup>		√	√			
INTTM03H <sup>Note 3</sup>	TMIF03H <sup>Note 3</sup>		TMMK03H <sup>Note 3</sup>		TMPR003H, TMPR103H <sup>Note 3</sup>		√	√			
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		√	√			
INTRTIT	RTITIF		RTITMK		RTITPR0, RTITPR1		√	√			
INTTM01	TMIF01	TMMK01	TMPR001, TMPR101	√	√						
INTTM02	TMIF02	IF1H	TMMK02	MK1H	TMPR002, TMPR102	PR01H, PR11H	√	√			
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		√	√			
INTAD	ADIF		ADMK		ADPR0, ADPR1		√	√			
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1		√	√			
INTIT	TMKAIF		TMKAMK		TMKAPR0, TMKAPR1		√	√			
INTKR	KRIF		KRMK		KRPR0, KRPR1		√	√			
INTST3	STIF3		STMK3		STPR03, STPR13		√	–			
INTSR3	SRIF3		SRMK3		SRPR03, SRPR13		√	–			
INTTKB20	TKBIF20		IF2L		TKBMK20		MK2L	TKBPR020, TKBPR120	PR02L, PR12L	√	√
INTTM04	TMIF04				TMMK04			TMPR004, TMPR104		√	√
INTTM05	TMIF05	TMMK05		TMPR005, TMPR105	√	√					
INTP6	PIF6	PMK6		PPR06, PPR16	√	√					
INTP7	PIF7	PMK7		PPR07, PPR17	√	√					
INTCMP0	CMPIF0	CMPMK0		CMPPR00, CMPPR10	√	√					
INTCMP1	CMPIF1	CMPMK1		CMPPR01, CMPPR11	√	√					

- Notes**
- 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 0 of the IF1L register is set to 1. Bit 0 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.
  - If one of the interrupt sources INTST1, INTCSI10, and INTIIC10 is generated, bit 1 of the IF1L register is set to 1. Bit 1 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.
  - Do not use a UART1 reception error interrupt and an interrupt of channel 3 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART1 reception error interrupt is not used (EOC03 = 0), UART1 and channel 3 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If one of the interrupt sources INTSRE1 and INTTM03H is generated, bit 3 of the IF1L register is set to 1. Bit 3 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

Table 19-2. Flags Corresponding to Interrupt Request Sources (3/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		80-pin	64-pin
		Register		Register		Register		
INTTM06	TMIF06	IF2H	TMMK06	MK2H	TMPR006, TMPR106	PR02H, PR12H	√	√
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		√	√
INTSRE3	SREIF3		SREMK3		SREPR03, SREPR13		√	–
INTMD	MDIF		MDMK		MDPR0, MDPR1		√	√
INTFL	FLIF		FLMK		FLPR0, FLPR1		√	√
INTDMA2	DMAIF2	IF3L	DMAMK2	MK3L	DMAPR02, DMAPR12	PR03L, PR13L	√	√
INTDMA3	DMAIF3		DMAMK3		DMAPR03, DMAPR13		√	√

**19.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L)**

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, IF2H, and IF3L 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 19-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L) (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	LVIIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SRIF0	TMIF00	STIF0 CSIIIF00 IICIF00	DMAIF1	DMAIF0	SREIF2	SRIF2	STIF2

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF01	0	RTITIF	IICAIF0	SREIF1 TMIF03H	SRIF1	STIF1 CSIIIF10 IICIF10	SREIF0 TMIF01H

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	SRIF3	STIF3	KRIF	TMKAIF	RTCIF	ADIF	TMIF03	TMIF02

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
IF2L	CMPIF1	CMPIF0	0	PIF7	PIF6	TMIF05	TMIF04	TKBIF20

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	2	<1>	<0>
IF2H	FLIF	0	MDIF	SREIF3	0	0	TMIF07	TMIF06

**Figure 19-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L) (2/2)**

Address: FFFD2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
IF3L	0	0	0	0	0	0	DMAIF3	DMAIF2

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

- Cautions**
1. The available bits differ depending on the product. For details about the bits available for each product, see Table 19-2. Be sure to clear bits that are not available to 0.
  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.

**19.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L)**

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt.

The MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, and MK3L 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 19-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L)**

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	TMMK00	STMK0 CSIMK00 IICMK00	DMAMK1	DMAMK0	SREMK2	SRMK2	STMK2

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK01	1	RTITMK	IICAMK0	SREMK1 TMMK03H	SRMK1	STMK1 CSIMK10 IICMK10	SREMK0 TMMK01H

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	SRMK3	STMK3	KRMK	TMKAMK	RTCMK	ADMK	TMMK03	TMMK02

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
MK2L	CMPMK1	CMPMK0	1	PMK7	PMK6	TMMK05	TMMK04	TKBMK20

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	6	<5>	<4>	3	2	<1>	<0>
MK2H	FLMK	1	MDMK	SREMK3	1	1	TMMK07	TMMK06

Address: FFFD6H After reset: FFH R/W

Symbol	7	6	5	4	3	2	<1>	<0>
MK3L	1	1	1	1	1	1	DMAMK3	DMAMK2

XXMKX	Interrupt servicing control							
0	Interrupt servicing enabled							
1	Interrupt servicing disabled							

**Caution** The available bits differ depending on the product. For details about the bits available for each product, see Table 19-2. Be sure to set bits that are not available to 1.



**19.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L)**

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, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and the PR13L 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 19-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (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	TMPR000	STPR00 CSIPR000 IICPR000	DMAPR01	DMAPR00	SREPR02	SRPR02	STPR02

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SRPR10	TMPR100	STPR10 CSIPR100 IICPR100	DMAPR11	DMAPR10	SREPR12	SRPR12	STPR12

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR001	1	RTITPR0	IICAPR00	SREPR01 TMPR003H	SRPR01	STPR01 CSIPR010 IICPR010	SREPR00 TMPR001H

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR101	1	RTITPR1	IICAPR10	SREPR11 TMPR103H	SRPR11	STPR11 CSIPR110 IICPR110	SREPR10 TMPR101H

**Figure 19-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (2/2)**

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	SRPR03	STPR03	KRPR0	TMKAPR0	RTCPR0	ADPR0	TMPR003	TMPR002

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	SRPR13	STPR13	KRPR1	TMKAPR1	RTCPR1	ADPR1	TMPR103	TMPR102

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
PR02L	CMPPR01	CMPPR00	1	PPR07	PPR06	TMPR005	TMPR004	TKBPR020

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
PR12L	CMPPR11	CMPPR10	1	PPR17	PPR16	TMPR105	TMPR104	TKBPR120

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	6	<5>	<4>	3	2	<1>	<0>
PR02H	FLPR0	1	MDPR0	SREPR03	1	1	TMPR007	TMPR006

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	6	<5>	<4>	3	2	<1>	<0>
PR12H	FLPR1	1	MDPR1	SREPR13	1	1	TMPR107	TMPR106

Address: FFFDAH After reset: FFH R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PR03L	1	1	1	1	1	1	DMAPR03	DMAPR02

Address: FFFDEH After reset: FFH R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PR13L	1	1	1	1	1	1	DMAPR13	DMAPR12

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 available bits differ depending on the product. For details about the bits available for each product, see Table 19-2. Be sure to set bits that are not available to 1.

**19.3.4 External interrupt rising edge enable register (EGP0), external interrupt falling edge enable register (EGN0)**

These registers specify the valid edge for INTP0 to INTP7.

The EGP0 and EGN0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Figure 19-5. Format of External Interrupt Rising Edge Enable Register (EGP0) and External Interrupt Falling Edge Enable Register (EGN0)**

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 7)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 19-3 shows the ports corresponding to the EGPn and EGNn bits.

**Table 19-3. Ports Corresponding to EGPn and EGNn bits**

Detection Enable Bit		Interrupt Request Signal
EGP0	EGN0	INTP0
EGP1	EGN1	INTP1
EGP2	EGN2	INTP2
EGP3	EGN3	INTP3
EGP4	EGN4	INTP4
EGP5	EGN5	INTP5
EGP6	EGN6	INTP6
EGP7	EGN7	INTP7

**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**
- For edge detection port, see 2.1 Port Function.
  - n = 0 to 7

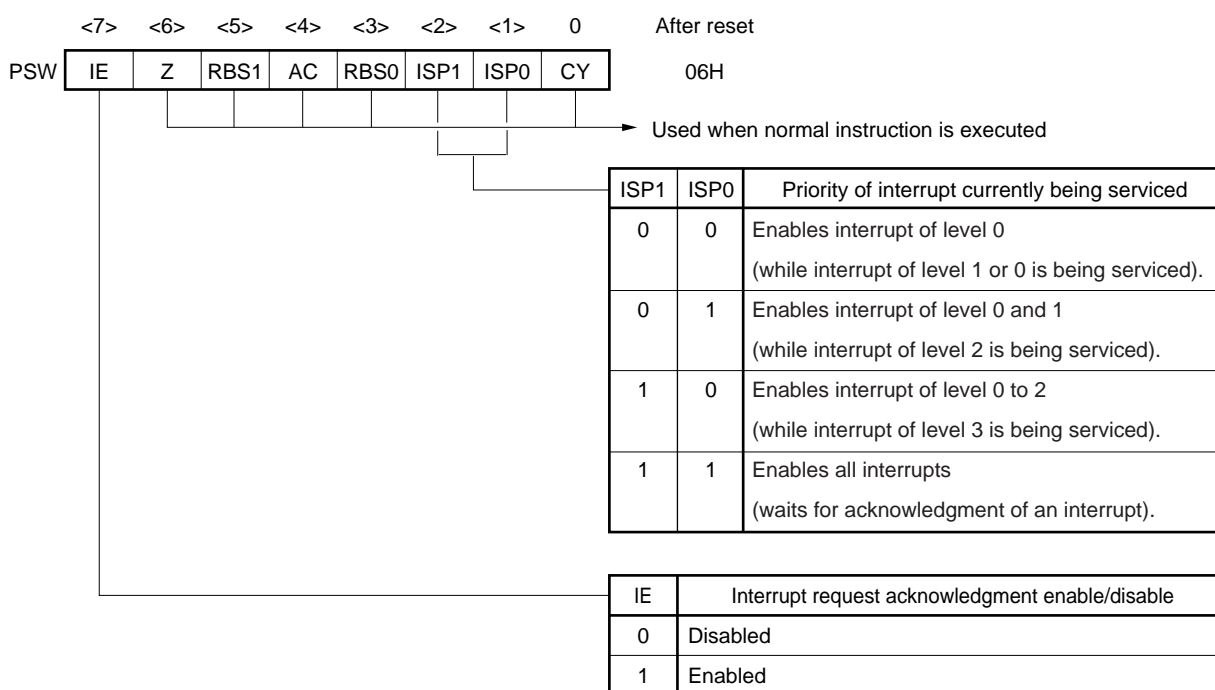
### 19.3.5 Program status word (PSW)

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 19-6. Configuration of Program Status Word



## 19.4 Interrupt Servicing Operations

### 19.4.1 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 19-4 below.

For the interrupt request acknowledgment timing, see **Figures 19-8 and 19-9**.

**Table 19-4. Time from Generation of Maskable Interrupt Until Servicing**

	Minimum Time	Maximum Time <sup>Note</sup>
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<sub>CLK</sub> (f<sub>CLK</sub>: 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.

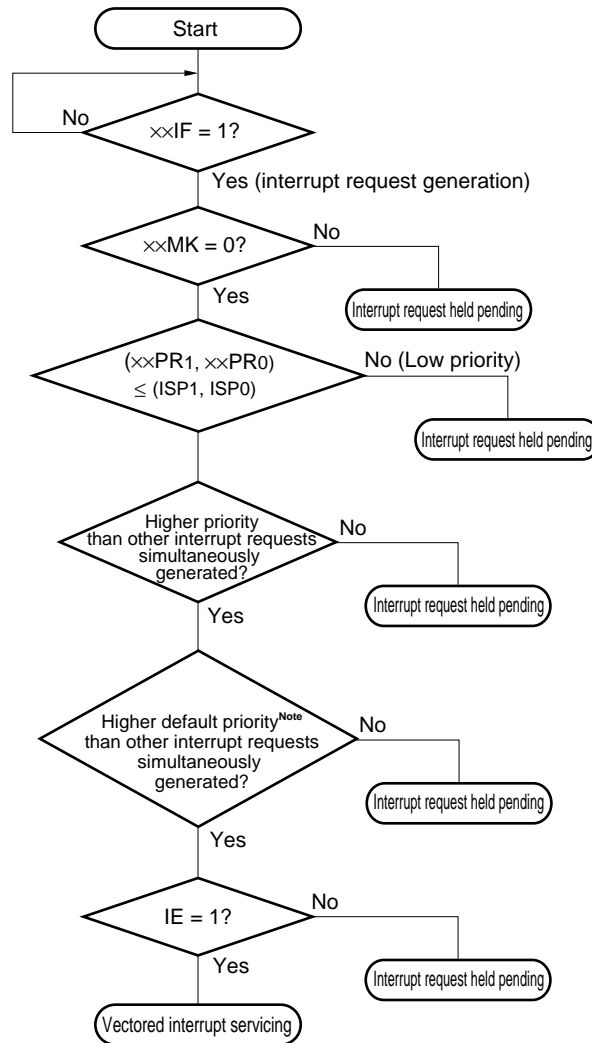
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 19-7 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.

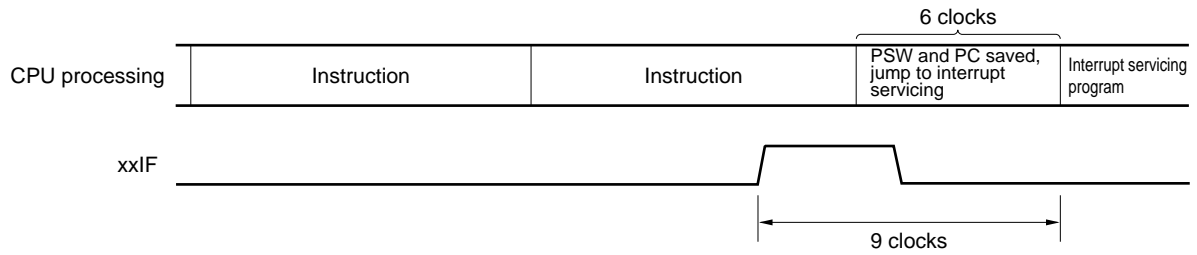
Figure 19-7. Interrupt Request Acknowledgment Processing Algorithm



- xxIF: Interrupt request flag
- xxMK: Interrupt mask flag
- xxPR0: Priority specification flag 0
- xxPR1: 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 19-6)

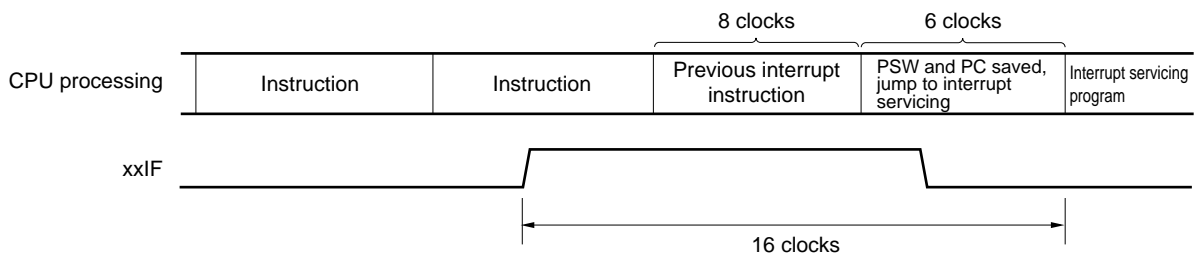
**Note** For the default priority, see Table 19-1 Interrupt Source List.

**Figure 19-8. Interrupt Request Acknowledgment Timing (Minimum Time)**



**Remark** 1 clock:  $1/f_{CLK}$  ( $f_{CLK}$ : CPU clock)

**Figure 19-9. Interrupt Request Acknowledgment Timing (Maximum Time)**



**Remark** 1 clock:  $1/f_{CLK}$  ( $f_{CLK}$ : CPU clock)

### 19.4.2 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.

### 19.4.3 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 19-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 19-10 shows multiple interrupt servicing examples.



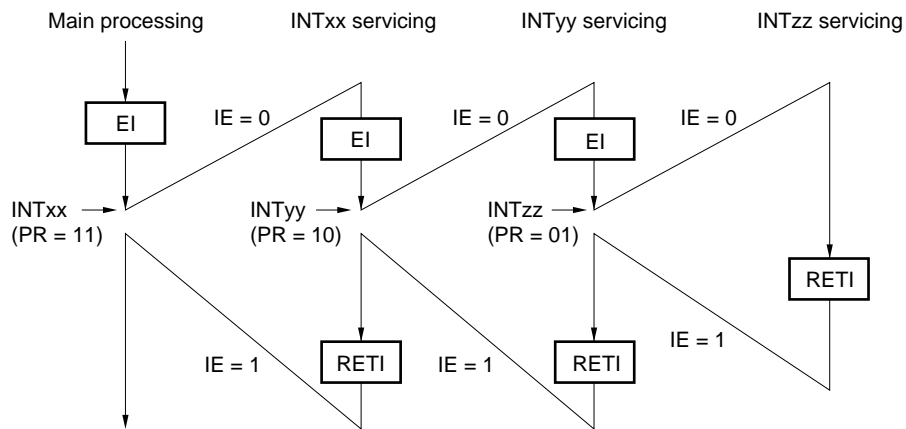
**Table 19-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing**

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**
- : Multiple interrupt servicing enabled
  - ×: Multiple interrupt servicing disabled
  - 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 (all interrupts are enabled).  
 IE = 0: Interrupt request acknowledgment is disabled.  
 IE = 1: Interrupt request acknowledgment is enabled.
  - PR is a flag contained in the PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L registers.  
 PR = 00: Specify level 0 with  $\text{xxPR1x} = 0, \text{xxPR0x} = 0$  (higher priority level)  
 PR = 01: Specify level 1 with  $\text{xxPR1x} = 0, \text{xxPR0x} = 1$   
 PR = 10: Specify level 2 with  $\text{xxPR1x} = 1, \text{xxPR0x} = 0$   
 PR = 11: Specify level 3 with  $\text{xxPR1x} = 1, \text{xxPR0x} = 1$  (lower priority level)

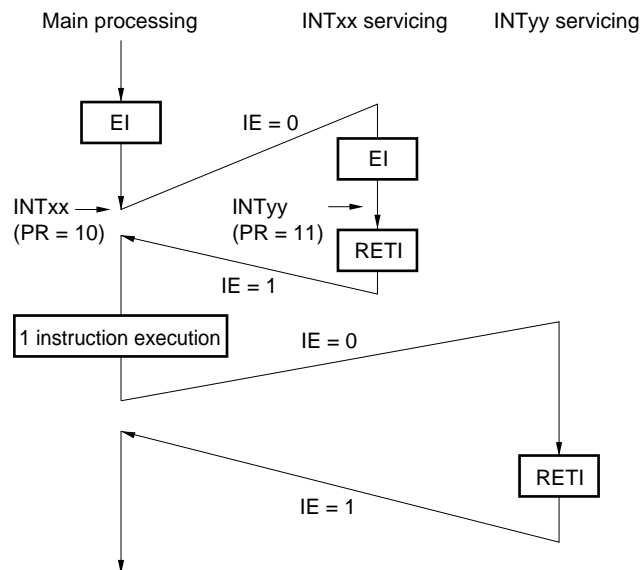
Figure 19-10. Examples of Multiple Interrupt Servicing (1/2)

**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.

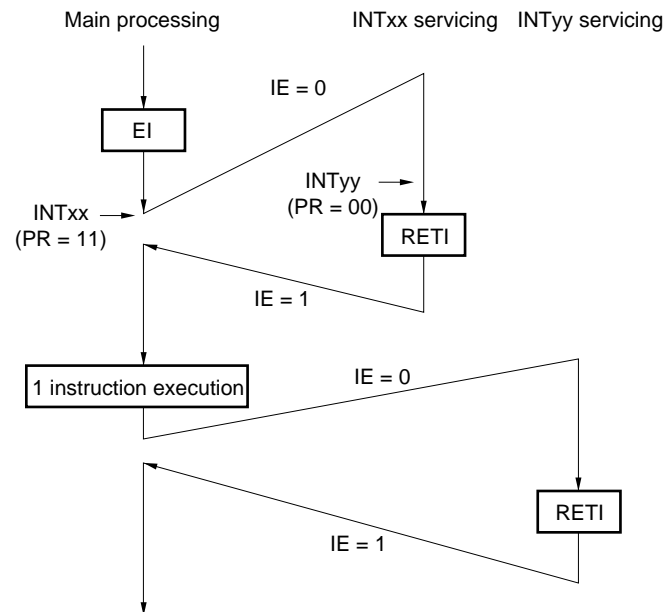
**Example 2. Multiple interrupt servicing does not occur due to priority control**



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  $\times\times PR1\times = 0, \times\times PR0\times = 0$  (higher priority level)
- PR = 01: Specify level 1 with  $\times\times PR1\times = 0, \times\times PR0\times = 1$
- PR = 10: Specify level 2 with  $\times\times PR1\times = 1, \times\times PR0\times = 0$
- PR = 11: Specify level 3 with  $\times\times PR1\times = 1, \times\times PR0\times = 1$  (lower priority level)
- IE = 0: Interrupt request acknowledgment is disabled
- IE = 1: Interrupt request acknowledgment is enabled.

Figure 19-10. 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  $\times\times PR1\times = 0$ ,  $\times\times PR0\times = 0$  (higher priority level)

PR = 01: Specify level 1 with  $\times\times PR1\times = 0$ ,  $\times\times PR0\times = 1$

PR = 10: Specify level 2 with  $\times\times PR1\times = 1$ ,  $\times\times PR0\times = 0$

PR = 11: Specify level 3 with  $\times\times PR1\times = 1$ ,  $\times\times PR0\times = 1$  (lower priority level)

IE = 0: Interrupt request acknowledgment is disabled

IE = 1: Interrupt request acknowledgment is enabled.

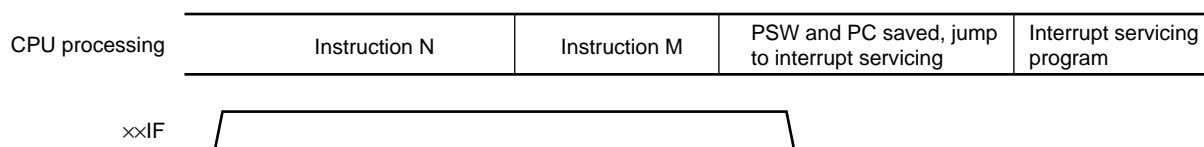
#### 19.4.4 Interrupt request hold

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
- MULHU
- MULH
- MACHU
- MACH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L registers

Figure 19-11 shows the timing at which interrupt requests are held pending.

**Figure 19-11. Interrupt Request Hold**



- Remarks**
1. Instruction N: Interrupt request hold instruction
  2. Instruction M: Instruction other than interrupt request hold instruction

## CHAPTER 20 KEY INTERRUPT FUNCTION

The number of key interrupt input channels differs, depending on the product.

	64-pin	80-pin
Key interrupt input channels	5 ch	8 ch

### 20.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by inputting a rising edge/falling edge to the key interrupt input pins (KR0 to KR7).

**Table 20-1. Assignment of Key Interrupt Detection Pins**

Key Interrupt Input Pins	Key Return Mode Register (KRM)
KR0	KRM00
KR1	KRM01
KR2	KRM02
KR3	KRM03
KR4	KRM04
KR5	KRM05
KR6	KRM06
KR7	KRM07

**Remark** KR1 to KR3: Available in the 80-pin products only.

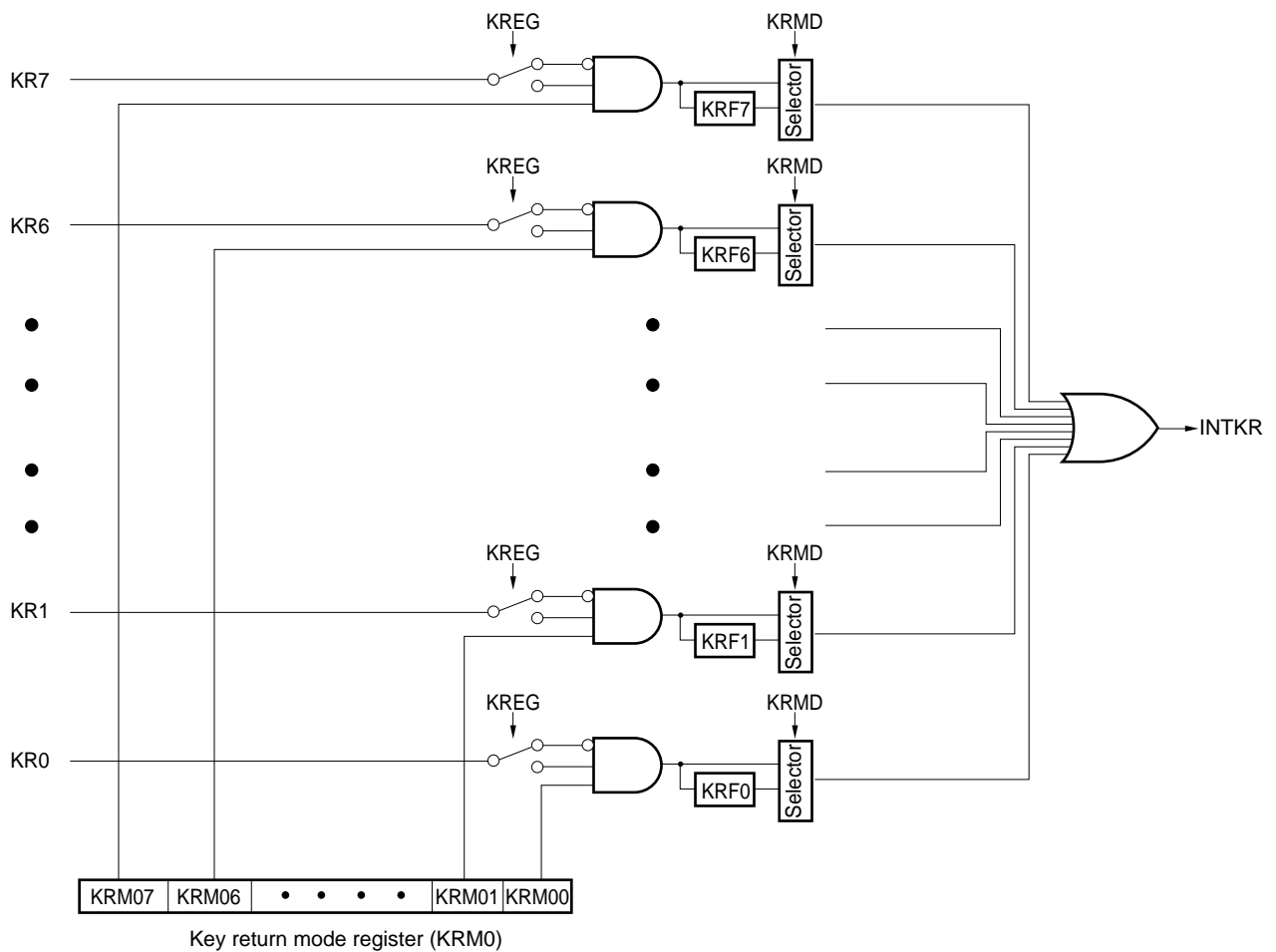
### 20.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

**Table 20-2. Configuration of Key Interrupt**

Item	Configuration
Input	KR0 to KR7
Control register	Key return control register (KRCTL) Key return mode register (KRM0) Key return flag register (KRF) Port mode register 7 (PM7) Port register 7 (P7)

**Figure 20-1. Block Diagram of Key Interrupt**



## 20.3 Register Controlling Key Interrupt

The key interrupt function is controlled by the following registers:

- Key return control register (KRCTL)
- Key return mode register (KRM0)
- Key return flag register (KRF)
- Port mode register 7 (PM7)
- Port register 7 (P7)

### 20.3.1 Key return control register (KRCTL)

This register controls the usage of the key return flags (KRF0 to KRF7) and sets the detection edge.

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

Reset signal generation clears this register to 00H.

**Figure 20-2. Format of Key Return Control Register (KRCTL)**

Address: FFF34H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRCTL	KRMD	0	0	0	0	0	0	KREG

KRMD	Usage of key return flags (KRF0 to KRF7)
0	Does not use key return flags
1	Uses key return flags

KREG	Selection of detection edge (KR0 to KR7)
0	Falling edge
1	Rising edge

### 20.3.2 Key return mode register (KRM0)

This register set the key interrupt mode.

The KRM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 20-3. Format of Key Return Mode Register (KRM0)**

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM0	KRM07	KRM06	KRM05	KRM04	KRM03 <sup>Note</sup>	KRM02 <sup>Note</sup>	KRM01 <sup>Note</sup>	KRM00

KRM0n	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

**Note** 80-pin products only.

- Cautions**
1. When selecting the falling edge detection, the on-chip pull-up resistor can be used by setting the corresponding bit (PU70 to PU77) in pull-up resistor option register 7 (PU7) to 1.
  2. An interrupt will be generated if the target bit of the KRM0 register is set while a low level (when the bit 0 (KREG) of the key return control register (KRCTL) is set to 0) or high level (when the bit 0 (KREG) of the key return control register (KRCTL) is set to 1) is being input to the key interrupt input pin.  
To ignore this interrupt, set the KRM0 register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input high-level width or low-level width (see 32.4 or 33.4 AC Characteristics).
  3. The bits not used in the key interrupt mode can be used as normal ports.



**20.3.3 Key return flag register (KRF)**

This register controls the key return flags (KRF0 to KRF7).  
 The KRF register can be set by an 8-bit memory manipulation instruction.  
 Reset signal generation clears this register to 00H.

**Figure 20-4. Format of Key Return Flag Register (KRF)**

Address: FFF35H After reset: 00H R/W<sup>Note 1</sup>

Symbol	7	6	5	4	3	2	1	0
KRF	KRF7	KRF6	KRF5	KRF4	KRF3 <sup>Note 2</sup>	KRF2 <sup>Note 2</sup>	KRF1 <sup>Note 2</sup>	KRF0

KRFn	Key interrupt flag (n = 0 to 7)
0	No key interrupt signal has been detected.
1	A key interrupt signal has been detected.

- Notes**
1. The bits in this register cannot be set to 1. To clear KRFn, write to the bits by using an 8-bit memory manipulation instruction in which 0 is written to the target bit and 1 is written to the other bits.
  2. 80-pin products only.

**Caution** When using the key return flag (KRM<sub>D</sub> = 1), if other KRM<sub>m</sub> is set to 1 before clearing KRFn (m ≠ n) that set to 1, the key interrupt is not generated. To acknowledge next key interrupt after key interrupt is generated, be sure to clear the corresponding KRFn.

**20.3.4 Port mode register 7 (PM7)**

Write 1 to the bit of port mode register (PM7) corresponding to each port when using P70/KR0 to P77/KR7 as a key input. 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 clears this register to FFH.

**Figure 20-5. Format of Port Mode Register 7 (PM7) (80-pin products)**

Address: FFF27H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	I/O mode selection for P7n/KRn pin (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

## CHAPTER 21 STANDBY FUNCTION

### 21.1 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 CSIp or UARTq 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 CSIp or UARTq 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 all these 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 CSIp, UARTq, or the A/D converter in the SNOOZE mode, set up serial standby control register m (SSCm) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 14.3 Registers Controlling Serial Array Unit and 12.3 Registers Used in 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 27 OPTION BYTE.

**Remark** p = 00; q = 0, 2; m = 0, 1

## 21.2 Registers Controlling Standby Function

The registers which control the standby function are described below.

- 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 12 A/D CONVERTER** and **CHAPTER 14 SERIAL ARRAY UNIT**.

## 21.3 Standby Function Operation

### 21.3.1 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 the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the HALT mode is not entered even if the HALT instruction is executed in such a situation.

**Table 21-1. Operating Statuses in HALT Mode (1/2)**

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 ( $f_{EX}$ )			
System clock		Clock supply to the CPU is stopped					
Main system clock	$f_{IH}$	Operation continues (cannot be stopped)	Operation disabled				
	$f_x$	Operation disabled	Operation continues (cannot be stopped)	Cannot operate			
	$f_{EX}$		Cannot operate	Operation continues (cannot be stopped)			
Subsystem clock	$f_{XT}$	Status before HALT mode was set is retained					
	$f_{EXS}$						
$f_{IL}$		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) <ul style="list-style-type: none"> <li>• WUTMMCK0 = 1: Oscillates</li> <li>• WUTMMCK0 = 0 and WDTON = 0: Stops</li> <li>• WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates</li> <li>• WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops</li> </ul>					
CPU		Operation stopped					
Code flash memory		Operation stopped					
Data flash memory							
RAM							
Port (latch)							
Timer array unit		Operable					
Timer KB20							
Real-time clock 2							
12-bit interval timer							
Watchdog timer		See <b>CHAPTER 11 WATCHDOG TIMER</b>					
Clock output/buzzer output		Operable					
A/D converter							
Comparator							
Serial array unit (SAU)							
Serial interface (IICA)							
LCD driver/controller					Operable (However, this depends on the status of the clock selected as the LCD source clock: operation is possible if the selected clock is operating, but operation will stop if the selected clock is stopped.)		
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	In the calculation of the RAM area, operable when DMA is executed only					
	General-purpose CRC						
RAM parity error detection function		Operable when DMA is executed only					
RAM guard function							
SFR guard function							
Illegal-memory access detection function							

(Remark is listed on the next page.)

**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 <sub>H</sub> :	High-speed on-chip oscillator clock	f <sub>EX</sub> :	External main system clock
f <sub>L</sub> :	Low-speed on-chip oscillator clock	f <sub>XT</sub> :	XT1 clock
f <sub>X</sub> :	X1 clock	f <sub>EXS</sub> :	External subsystem clock

**Table 21-1. Operating Statuses in HALT Mode (2/2)**

HALT Mode Setting Item		When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock	
		When CPU Is Operating on XT1 Clock (f <sub>XT</sub> )	When CPU Is Operating on External Subsystem Clock (f <sub>EXS</sub> )
System clock		Clock supply to the CPU is stopped	
Main system clock	f <sub>IH</sub>	Operation disabled	
	f <sub>X</sub>		
	f <sub>EX</sub>		
Subsystem clock	f <sub>XT</sub>	Operation continues (cannot be stopped)	Cannot operate
	f <sub>EXS</sub>	Cannot operate	Operation continues (cannot be stopped)
f <sub>IL</sub>		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) <ul style="list-style-type: none"> <li>• WUTMMCK0 = 1: Oscillates</li> <li>• WUTMMCK0 = 0 and WDTON = 0: Stops</li> <li>• WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates</li> <li>• WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops</li> </ul>	
CPU		Operation stopped	
Code flash memory			
Data flash memory			
RAM			
Port (latch)			
Timer array unit		Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
Timer KB20		Operable	
Real-time clock 2			
12-bit interval timer			
Watchdog timer		See <b>CHAPTER 11 WATCHDOG TIMER</b>	
Clock output/buzzer output		Operable	
A/D converter		Operation disabled	
Comparator		Operable when external input (IVREFn) is selected for comparator reference voltage.	
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	
LCD driver/controller		Operable (However, this depends on the status of the clock selected as the LCD source clock: operation is possible if the selected clock is operating, but operation will stop if the selected clock is stopped.)	
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	
	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 is listed on the next page.)

**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 <sub>H</sub> : High-speed on-chip oscillator clock	f <sub>EX</sub> : External main system clock
f <sub>L</sub> : Low-speed on-chip oscillator clock	f <sub>XT</sub> : XT1 clock
f <sub>X</sub> : X1 clock	f <sub>EXS</sub> : External subsystem clock

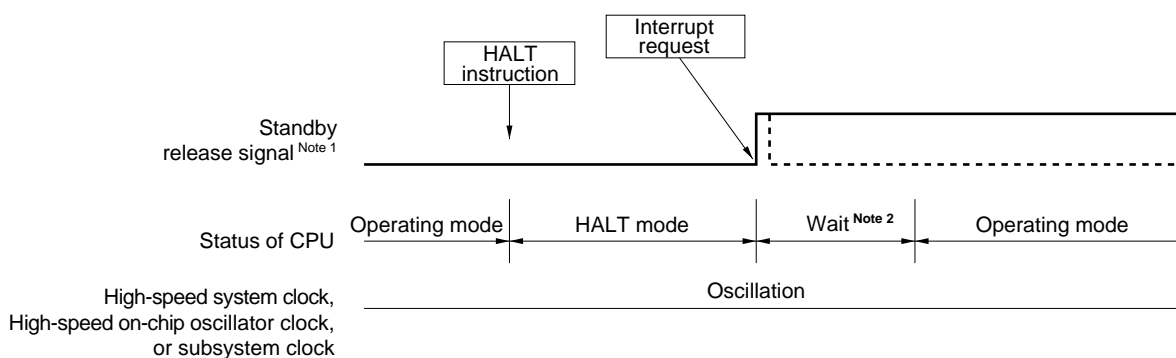
**(2) HALT mode release**

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 21-1. HALT Mode Release by Interrupt Request Generation**



**Notes 1.** For details of the standby release signal, see **Figure 19-1**

**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

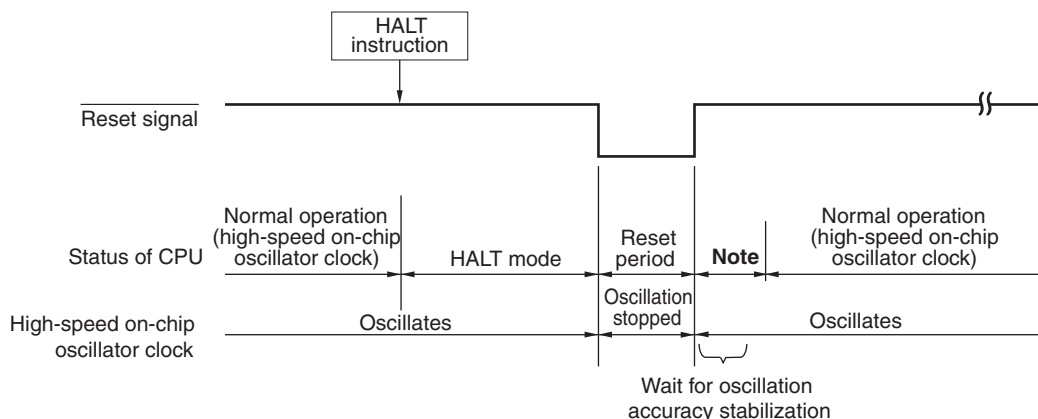
**Remark** The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

**(b) Release by reset signal generation**

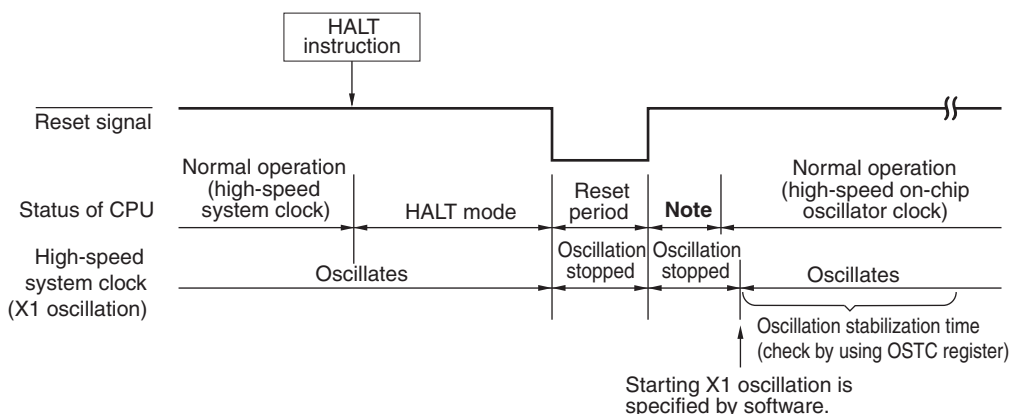
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

**Figure 21-2. HALT Mode Release by Reset (1/2)**

**(1) When high-speed on-chip oscillator clock is used as CPU clock**



**(2) When high-speed system clock is used as CPU clock**

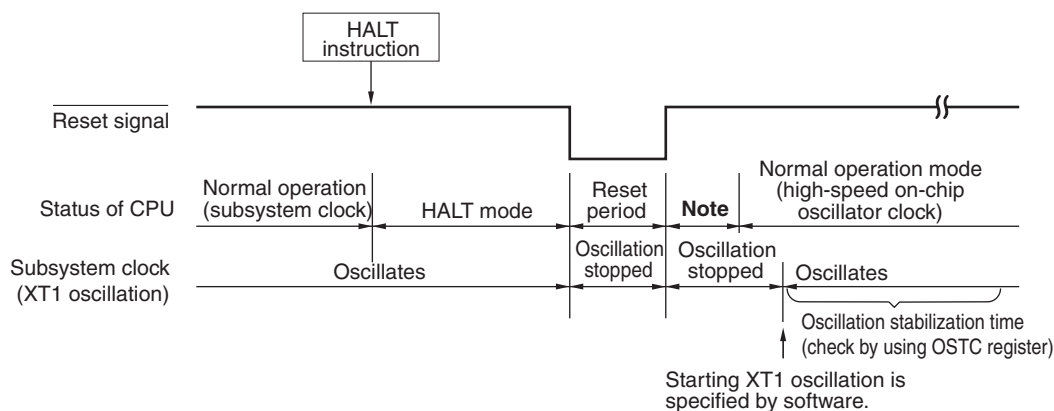


**Note** For the reset processing time, see **CHAPTER 22 RESET FUNCTION**. For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 23 POWER-ON-RESET CIRCUIT**.



Figure 21-2. HALT Mode Release by Reset (2/2)

(3) When subsystem clock is used as CPU clock



**Note** For the reset processing time, see **CHAPTER 22 RESET FUNCTION**. For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 23 POWER-ON-RESET CIRCUIT**.

21.3.2 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.

**Caution** Because the interrupt request signal is used to clear the STOP mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the STOP mode is immediately cleared if set when the STOP instruction is executed in such a situation. Accordingly, once the STOP instruction is executed, the system returns to its normal operating mode after the elapse of release time from the STOP mode.

The operating statuses in the STOP mode are shown below.

Table 21-2. Operating Statuses in STOP Mode

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 ( $f_{EX}$ )
System clock		Clock supply to the CPU is stopped		
Main system clock	$f_{IH}$	Stopped		
	$f_x$			
	$f_{EX}$			
Subsystem clock	$f_{XT}$	Status before STOP mode was set is retained		
	$f_{EXS}$			
$f_{IL}$		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) <ul style="list-style-type: none"> <li>• WUTMMCK0 = 1: Oscillates</li> <li>• WUTMMCK0 = 0 and WDTON = 0: Stops</li> <li>• WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates</li> <li>• WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops</li> </ul>		
CPU		Operation stopped		
Code flash memory				
Data flash memory		Operation stopped		
RAM		Operation stopped		
Port (latch)		Status before STOP mode was set is retained		
Timer array unit		Operation disabled		
Timer KB20				
Real-time clock 2		Operable		
12-bit interval timer				
Watchdog timer		See <b>CHAPTER 11 WATCHDOG TIMER</b>		
Clock output/buzzer output		Operable only when subsystem clock is selected as the count clock		
A/D converter		Wakeup operation is enabled (switching to the SNOOZE mode)		
Comparator		Operable (only when digital filter is not used and external input (IVREFn) is selected for comparator reference voltage)		
Serial array unit (SAU)		Wakeup operation is enabled only for CSIp and UARTq (switching to the SNOOZE mode) Operation is disabled for anything other than CSIp and UARTq		
Serial interface (IICA)		Wakeup by address match operable		
LCD driver/controller		Operable (However, this depends on the status of the clock selected as the LCD source clock: operation is possible if the selected clock is operating, but operation will stop if the selected clock is stopped.)		
Multiplier and divider/multiply-accumulator		Operation disabled		
DMA controller				
Power-on-reset function		Operable		
Voltage detection function				
External interrupt				
Key interrupt function				
Key interrupt function				
CRC operation function	High-speed CRC	Operation stopped		
	General-purpose CRC			
RAM parity error detection function				
RAM guard function				
SFR guard function				
Illegal-memory access detection function				

(Remarks are listed on the next page.)

- Remarks 1.** Operation stopped: Operation is automatically stopped before switching to the STOP mode.  
 Operation disabled: Operation is stopped before switching to the STOP mode.  
 $f_{IH}$ : High-speed on-chip oscillator clock     $f_{IL}$ : Low-speed on-chip oscillator clock  
 $f_x$ : X1 clock     $f_{EX}$ : External main system clock  
 $f_{XT}$ : XT1 clock     $f_{EXS}$ : External subsystem clock
- 2.**  $p = 00$ ;  $q = 0, 2$

## (2) STOP mode release

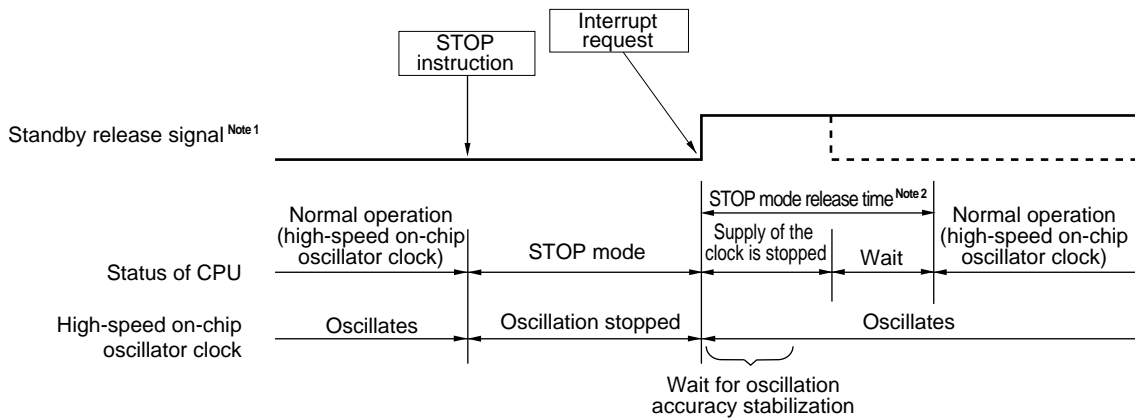
The STOP mode can be released by the following two sources.

### (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 21-3. STOP Mode Release by Interrupt Request Generation (1/3)**

#### (1) When high-speed on-chip oscillator clock is used as CPU clock



**Notes 1.** For details of the standby release signal, see **Figure 19-1**.

**2.** STOP mode release time

Supply of the clock is stopped:

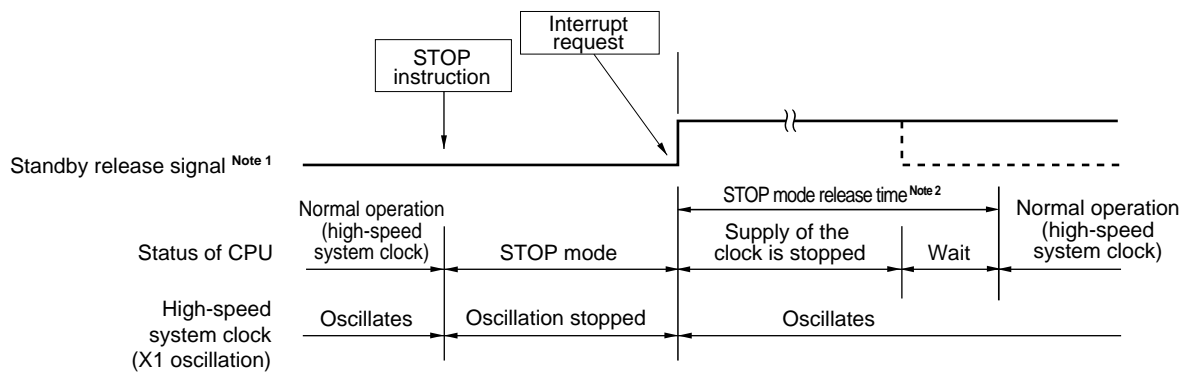
- $FRQSEL4 = 0$ :  $18 \mu s$  to “whichever is longer  $65 \mu s$  or the oscillation stabilization time (set by OSTS)”
- $FRQSEL4 = 1$ :  $18 \mu s$  to “whichever is longer  $80 \mu s$  or the oscillation stabilization time (set by OSTS)”

Wait

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

**Remarks 1.** The clock supply stop time varies depending on the temperature conditions and STOP mode period.

**2.** The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

**Figure 21-3. STOP Mode Release by Interrupt Request Generation (2/3)****(2) When high-speed system clock (X1 oscillation) is used as CPU clock**

**Notes** 1. For details of the standby release signal, see **Figure 19-1**

2. STOP mode release time

Supply of the clock is stopped:

- FRQSEL4 = 0: 18  $\mu$ s to “whichever is longer 65  $\mu$ s or the oscillation stabilization time (set by OSTs)”
- FRQSEL4 = 1: 18  $\mu$ s to “whichever is longer 80  $\mu$ s or 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

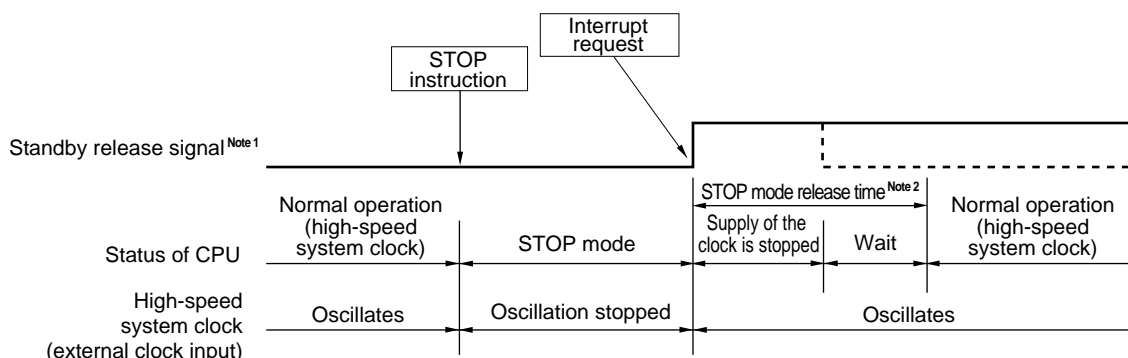
**Caution** To reduce the oscillation stabilization time after release from the STOP mode while CPU operates based on the high-speed system clock (X1 oscillation), switch the clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.

**Remarks** 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 21-3. STOP Mode Release by Interrupt Request Generation (3/3)

(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 19-1

2. STOP mode release time

Supply of the clock is stopped:

- FRQSEL4 = 0: 18  $\mu$ s to “whichever is longer 65  $\mu$ s or the oscillation stabilization time (set by OSTS)”
- FRQSEL4 = 1: 18  $\mu$ s to “whichever is longer 80  $\mu$ s or the oscillation stabilization time (set by OSTS)”

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 clock supply stop time varies depending on the temperature conditions and STOP mode period.

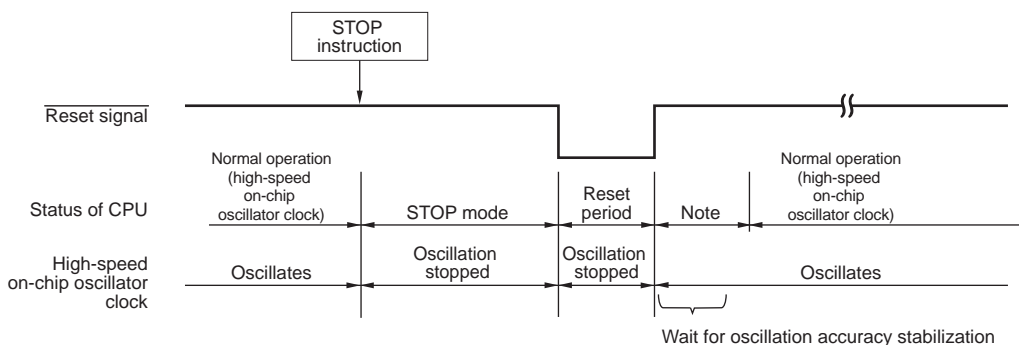
2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

**(b) Release by reset signal generation**

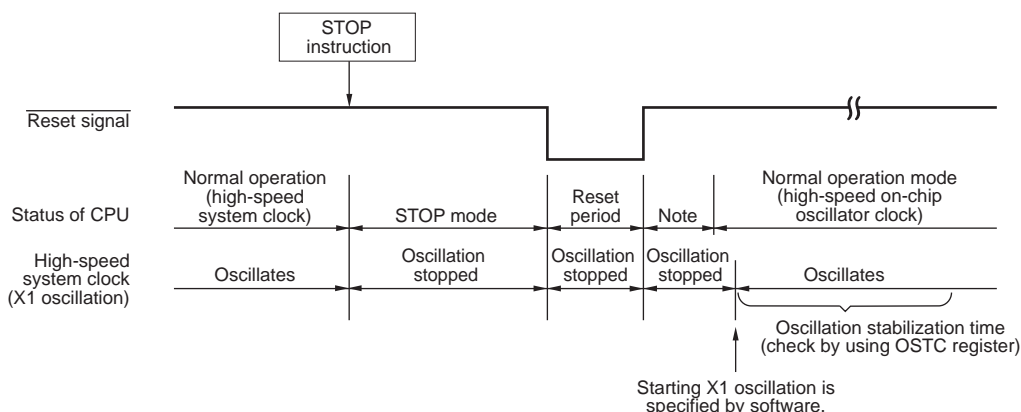
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

**Figure 21-4. 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 22 RESET FUNCTION**. For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 23 POWER-ON-RESET CIRCUIT**.

### 21.3.3 SNOOZE mode

#### (1) SNOOZE mode setting and operating statuses

The SNOOZE mode can only be specified for CSIp, UARTq, 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 CSIp or UARTq in the SNOOZE mode, set the SWCm bit of serial standby control register m (SSCm) to 1 immediately before switching to the STOP mode. For details, see **14.3 Registers Controlling Serial Array Unit**.

When using the A/D converter in the SNOOZE mode, set the AWC bit of A/D converter mode register 2 (ADM2) to 1 immediately before switching to the STOP mode. For details, see **12.3 Registers Used in A/D Converter**.

**Remark** p = 00; q = 0, 2; m = 0, 1

The following time is required for mode transition.

Transition time from STOP mode to SNOOZE mode:

FRQSEL4 = 0: 18 to 65  $\mu$ s

FRQSEL4 = 1: 18 to 80  $\mu$ s

Transition time from SNOOZE mode to normal operation:

- When vectored interrupt servicing is carried out:
  - HS (High-speed main) mode : 4.99 to 9.44  $\mu$ s + 7 clocks
  - LS (Low-speed main) mode : 1.10 to 5.08  $\mu$ s + 7 clocks
  - LV (Low-voltage main) mode : 16.58 to 25.40  $\mu$ s + 7 clocks
- When vectored interrupt servicing is not carried out:
  - HS (High-speed main) mode : 4.99 to 9.44  $\mu$ s + 1 clock
  - LS (Low-speed main) mode : 1.10 to 5.08  $\mu$ s + 1 clock
  - LV (Low-voltage main) mode : 16.58 to 25.40  $\mu$ s + 1 clock

The operating statuses in the SNOOZE mode are shown below.

**Table 21-3. Operating Statuses in SNOOZE Mode**

STOP Mode Setting		When Inputting CSIp/UARTq 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	$f_{IH}$	Operation started	
	$f_X$	Stopped	
	$f_{EX}$		
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) <ul style="list-style-type: none"> <li>• WUTMMCK0 = 1: Oscillates</li> <li>• WUTMMCK0 = 0 and WDTON = 0: Stops</li> <li>• WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates</li> <li>• WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops</li> </ul>	
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	
Timer KB20			
Real-time clock 2		Operable	
12-bit interval timer			
Watchdog timer		See <b>CHAPTER 11 WATCHDOG TIMER</b>	
Clock output/buzzer output		Operable only when subsystem clock is selected as the count clock	
A/D converter		Operable	
Comparator		Operable (when digital filter is not used)	
Serial array unit (SAU)		Operable only CSIp and UARTq only. Operation disabled other than CSIp and UARTq.	
Serial interface (IICA)		Operation disabled	
LCD driver/controller		Operable (However, this depends on the status of the clock selected as the LCD source clock: operation is possible if the selected clock is operating, but operation will stop if the selected clock is stopped.)	
Multiplier and divider/multiply-accumulator		Operation disabled	
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			

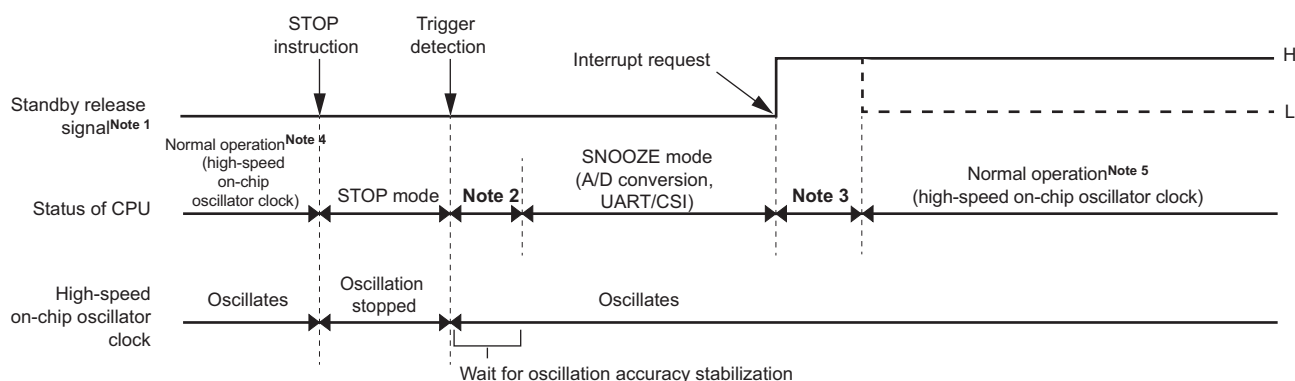
(Remark is listed on the next page.)



- Remarks 1.** Operation stopped: Operation is automatically stopped before switching to the SNOOZE mode.  
 Operation disabled: Operation is stopped before switching to the SNOOZE mode.  
 f<sub>IH</sub>: High-speed on-chip oscillator clock    f<sub>IL</sub>: Low-speed on-chip oscillator clock  
 f<sub>X</sub>: X1 clock    f<sub>EX</sub>: External main system clock  
 f<sub>XT</sub>: XT1 clock    f<sub>EXS</sub>: External subsystem clock
- 2.** p = 00; q = 0, 2

**(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode**

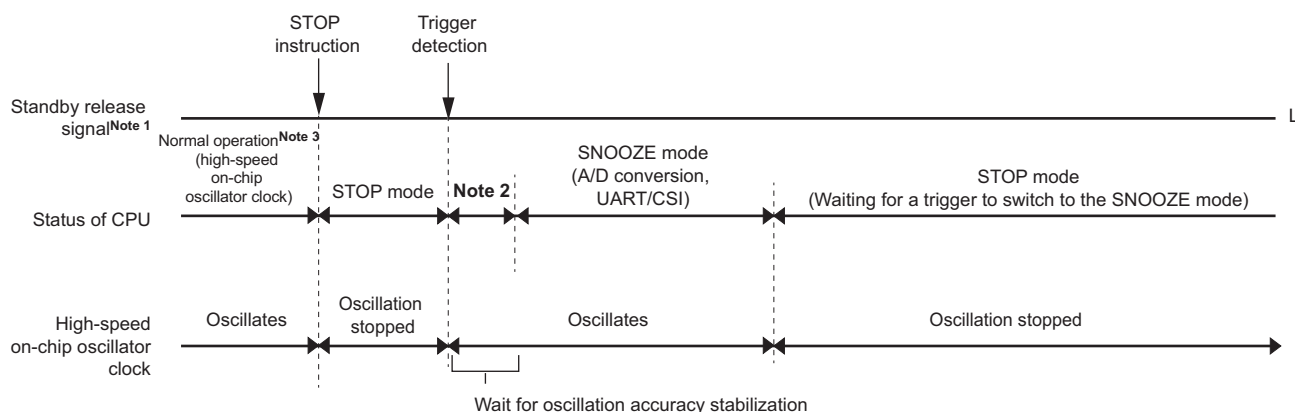
**Figure 21-5. When the Interrupt Request Signal is Generated in the SNOOZE Mode**



- Notes**
1. For details of the standby release signal, see **Figure 19-1**.
  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 SWC = 1) immediately before switching to the STOP mode.
  5. Be sure to release the SNOOZE mode (AWC = 0 or SWC = 0) immediately after return to the normal operation.

**(3) Timing diagram when the interrupt request signal is not generated in the SNOOZE mode**

**Figure 21-6. When the Interrupt Request Signal is not Generated in the SNOOZE Mode**



- Notes**
1. For details of the standby release signal, see **Figure 19-1**.
  2. Transition time from STOP mode to SNOOZE mode
  3. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.

**Remark** For details of the SNOOZE mode function, see **CHAPTER 12 A/D CONVERTER** and **CHAPTER 14 SERIAL ARRAY UNIT**.

## CHAPTER 22 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<sup>Note</sup>
- (6) Internal reset by RAM parity error
- (7) Internal reset by illegal-memory access

<R> 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<sup>Note</sup>, RAM parity error or illegal-memory access, and each item of hardware is set to the status shown in Table 22-1.

**Note** This reset occurs when instruction code FFH is executed.

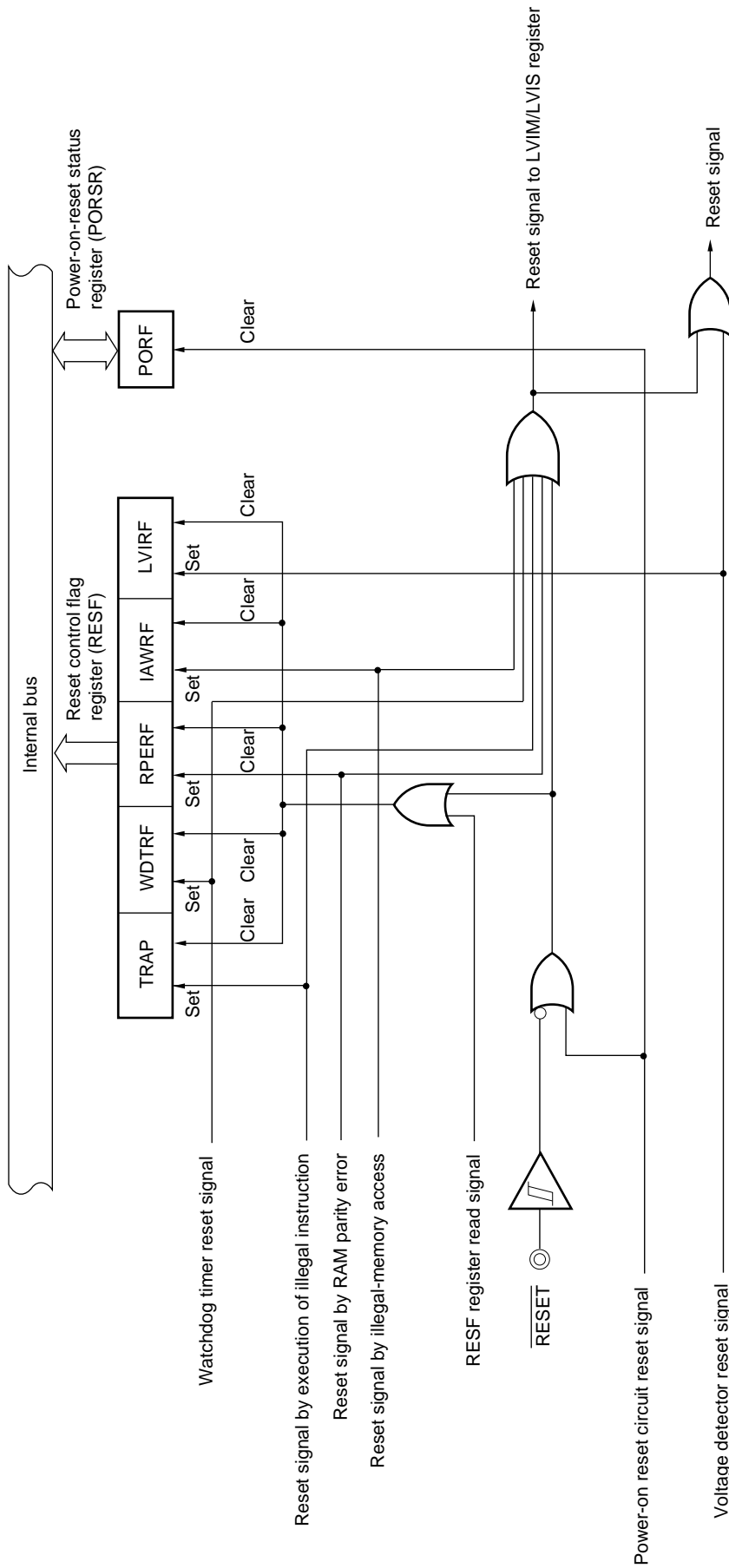
This reset does not occur during emulation using an in-circuit emulator or an on-chip debugging emulator.

**Cautions** 1. For an external reset, input a low level for 10  $\mu\text{s}$  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  $\mu\text{s}$  or more within the operating voltage range shown in 32.4 or 33.4 AC Characteristics, and then input a high level to the pin.

2. During reset input, the X1 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. When reset is effected, port pins become high-impedance, because each SFR and 2nd SFR are initialized.
  - 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 on-chip pull-up resistor).
  - Ports other than P40: High-impedance during the reset period or after receiving a reset signal.

Figure 22-1. Block Diagram of Reset Function



**Caution** An LVD circuit internal reset does not reset the LVD circuit.

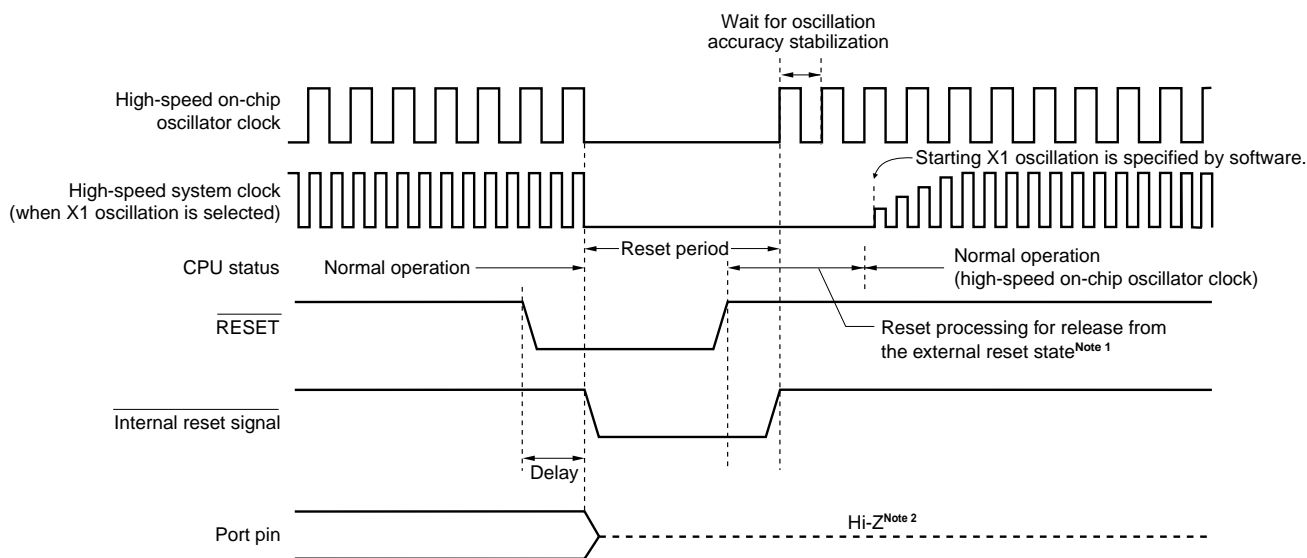
**Remarks 1.** LVIM: Voltage detection register

**2.** LVIS: Voltage detection level register

### 22.1 Timing of Reset Operation

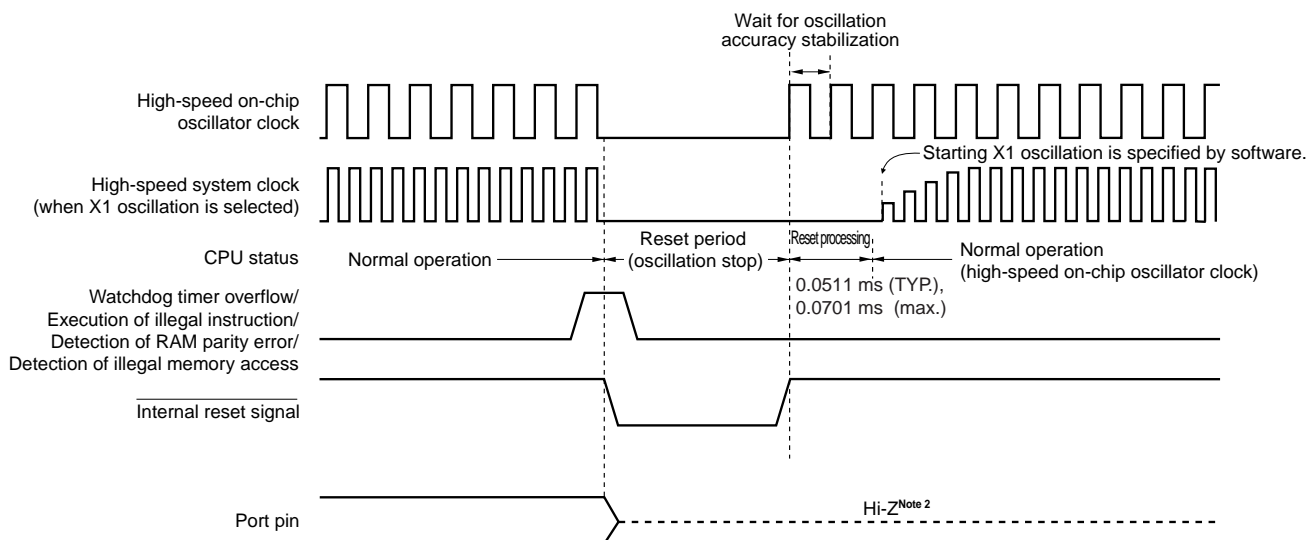
This LSI is reset by input of the low level on the  $\overline{\text{RESET}}$  pin and released from the reset state by input of the high level on the  $\overline{\text{RESET}}$  pin. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 22-2. 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 22-3. Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal Memory Access Overflow



(Notes are listed on the next page.)

**Notes 1.** 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.

**2.** 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 23 POWER-ON-RESET CIRCUIT** or **CHAPTER 24 VOLTAGE DETECTOR**.

**Remark**  $V_{POR}$ : POR power supply rise detection voltage

$V_{LVD}$ : LVD detection voltage

## 22.2 States of Operation During Reset Periods

Table 22-1 shows the states of operation during reset periods. Table 22-2 shows the states of the hardware after receiving a reset signal.

**Table 22-1. Operation Statuses During Reset Period**

Item		During Reset Period	
System clock		Clock supply to the CPU is stopped.	
Main system clock	$f_{IH}$	Operation stopped	
	$f_X$	Operation stopped (the X1 and X2 pins are input port mode)	
	$f_{EX}$	Clock input invalid (the pin is input port mode)	
Subsystem clock	$f_{XT}$	Operation possible (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		Operation stopped	
Code flash memory		Operation stopped	
Data flash memory		Operation stopped	
RAM		Operation stopped	
Port (latch)		High impedance <sup>Note</sup>	
Timer array unit		Operation stopped	
16-bit timer KB20			
Real-time clock 2		During a reset other than the POR reset: Operation possible During a POR reset: Calendar operation possible; operation of the RTCC0, RTCC1, and SUBCUD registers stops.	
12-bit interval timer		Operation stopped	
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Comparator			
Serial array unit (SAU)			
Serial interface (IICA)			
LCD controller/driver			
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** are listed on the next page.)

**Note** High-impedance during the external reset period or reset period by the POR. High level during other types of reset (connected to the on-chip pull-up resistor).

**Remark** f<sub>H</sub>: High-speed on-chip oscillator clock

f<sub>X</sub>: X1 oscillation clock

f<sub>EX</sub>: External main system clock

f<sub>XT</sub>: XT1 oscillation clock

f<sub>EXS</sub>: External subsystem clock

f<sub>L</sub>: Low-speed on-chip oscillator clock

**Table 22-2. State of Hardware After Receiving a Reset Signal**

Hardware		After Reset Acknowledgment <sup>Note</sup>
<R>	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** For the state of the special function register (SFR) after receiving a reset signal, see **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

## 22.3 Register for Confirming Reset Source

### 22.3.1 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 22-4. Format of Reset Control Flag Register (RESF)**

Address: FFFA8H After reset: Undefined<sup>Note 1</sup> 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 <sup>Note 2</sup>
0	No internal reset request has been generated, or the RESF register has been cleared.
1	An internal reset request has been generated.

WDTRF	Internal reset request by watchdog timer (WDT)
0	No internal reset request has been generated, or the RESF register has been cleared.
1	An internal reset request has been generated.

RPERF	Internal reset request by RAM parity
0	No internal reset request has been generated, or the RESF register has been cleared.
1	An internal reset request has been generated.

IAWRF	Internal reset request by illegal-memory access
0	No internal reset request has been generated, or the RESF register has been cleared.
1	An internal reset request has been generated.

LVIRF	Internal reset request by voltage detector (LVD)
0	No internal reset request has been generated, or the RESF register has been cleared.
1	An internal reset request has been generated.

- Notes**
- The value after reset varies depending on the reset source. See **Table 22-3**.
  - This reset occurs when instruction code FFH is executed.  
This reset does not occur during emulation using an in-circuit emulator or an on-chip debugging emulator.

- Cautions**
- Do not read data by a 1-bit memory manipulation instruction.
  - 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 25.3.3 RAM parity error detection function.



The status of the RESF register when a reset request is generated is shown in Table 22-3.

**Table 22-3. RESF Register Status When Reset Request Is Generated**

Reset Source Flag	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 bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held	Held
WDTRF bit			Held	Set (1)			
RPERF bit				Held	Set (1)		
IAWRF bit			Held		Set (1)		
LVIRF bit				Held	Set (1)		

The RESF register is automatically cleared when it is read by an 8-bit memory manipulation instruction. Figure 22-6 shows the procedure for checking a reset source.

**22.3.2 Power-on-reset status register (PORSR)**

The PORSR register is used to check the occurrence of a power-on reset. Writing 1 to bit 0 (PORF) of the PORSR register enables this function. Writing 0 disables this function. Write 1 to the PORF bit in advance to enable checking of the occurrence of a power-on reset. The PORSR register can be set by an 8-bit memory manipulation instruction. Power-on reset signal generation clears this register to 00H.

- Cautions 1. The PORSR register is reset only by a power-on reset; it retains the value when a reset caused by another factor occurs.**
- 2. If the PORF bit is set to 1, it guarantees that no power-on reset has occurred, but it does not guarantee that the RAM value is retained.**

**Figure 22-5. Format of Power-on-Reset Status Register (PORSR)**

Address: F00F9H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
PORSR	0	0	0	0	0	0	0	PORF

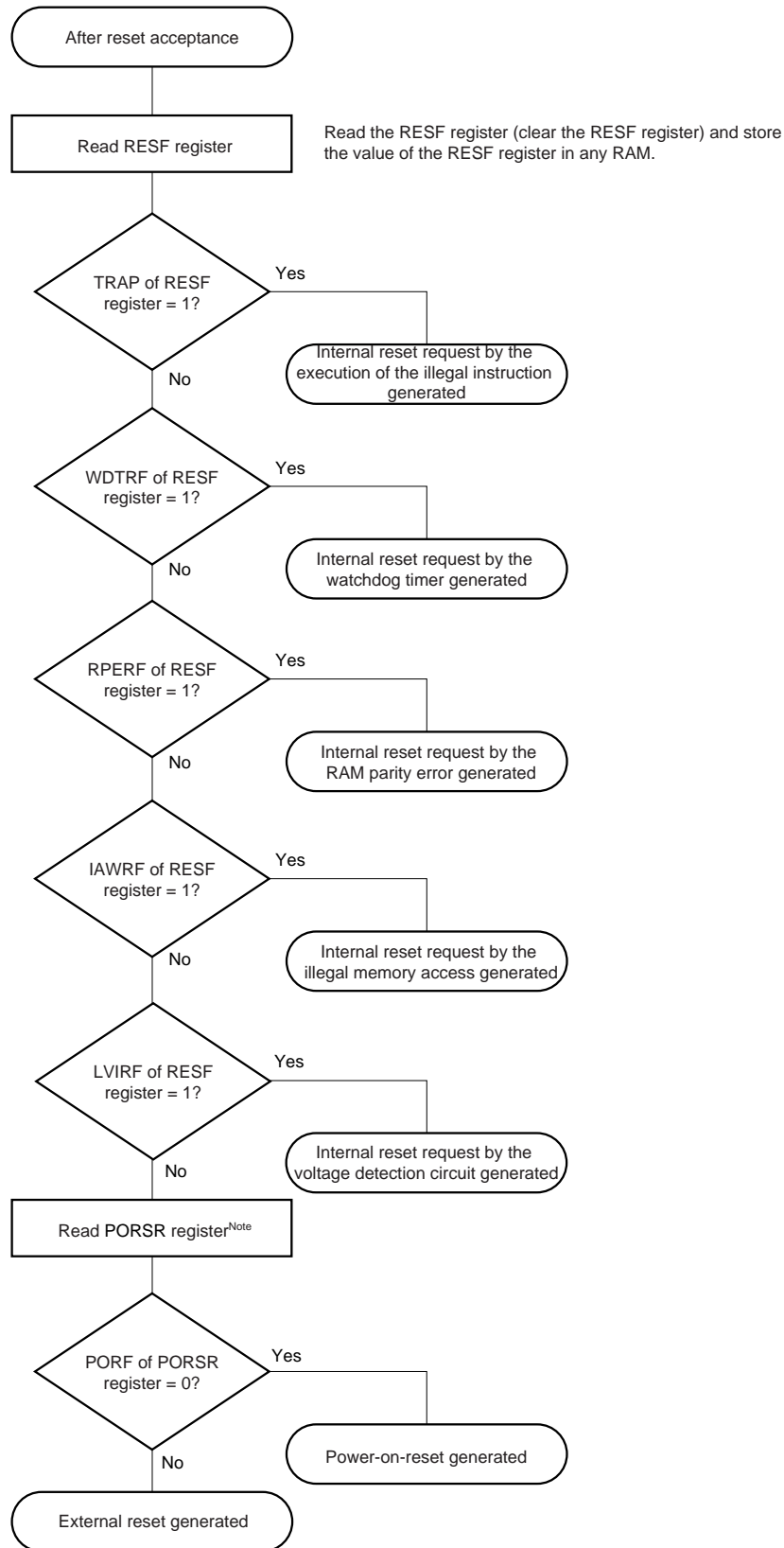
  

PORF	Checking occurrence of power-on reset
0	A value 1 has not been written, or a power-on reset has occurred.
1	No power-on reset has occurred.

Figure 22-6 shows the procedure for checking a reset source.

<R>

**Figure 22-6. Example of Procedure for Checking Reset Source**



**Note** Writing “1” to bit 0 (PORF) of the PORSR register before receiving a reset signal.

The flow described above is an example of the procedure for checking.

## CHAPTER 23 POWER-ON-RESET CIRCUIT

### 23.1 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 the reset state must be retained until the operating voltage becomes in the range defined in **32.4** or **33.4 AC Characteristics**. This is done by utilizing the voltage detector or controlling the externally input reset signal.
- Compares supply voltage ( $V_{DD}$ ) and detection voltage ( $V_{PDR}$ ), generates internal reset signal when  $V_{DD} < V_{PDR}$ . Note that, after power is supplied, 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 operation voltage falls below the range defined in **32.4** or **33.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

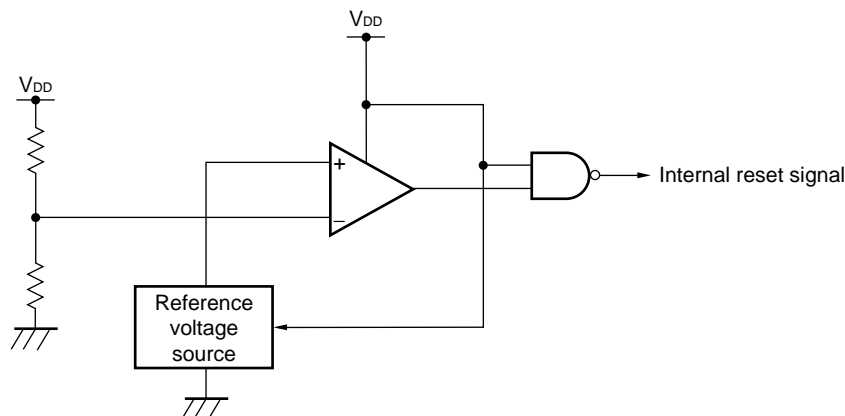
**Caution** If an internal reset signal is generated in the power-on-reset circuit, the reset control flag register (RESF) and power-on-reset status register (PORSR) are cleared.

- Remarks**
1. The RL78/L13 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 22 RESET FUNCTION**.
  2. Whether an internal reset has been generated by the power-on reset circuit can be checked by using the power-on-reset status register (PORSR). For details of the PORSR register, see **CHAPTER 22 RESET FUNCTION**.

### 23.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 23-1.

Figure 23-1. Block Diagram of Power-on-reset Circuit

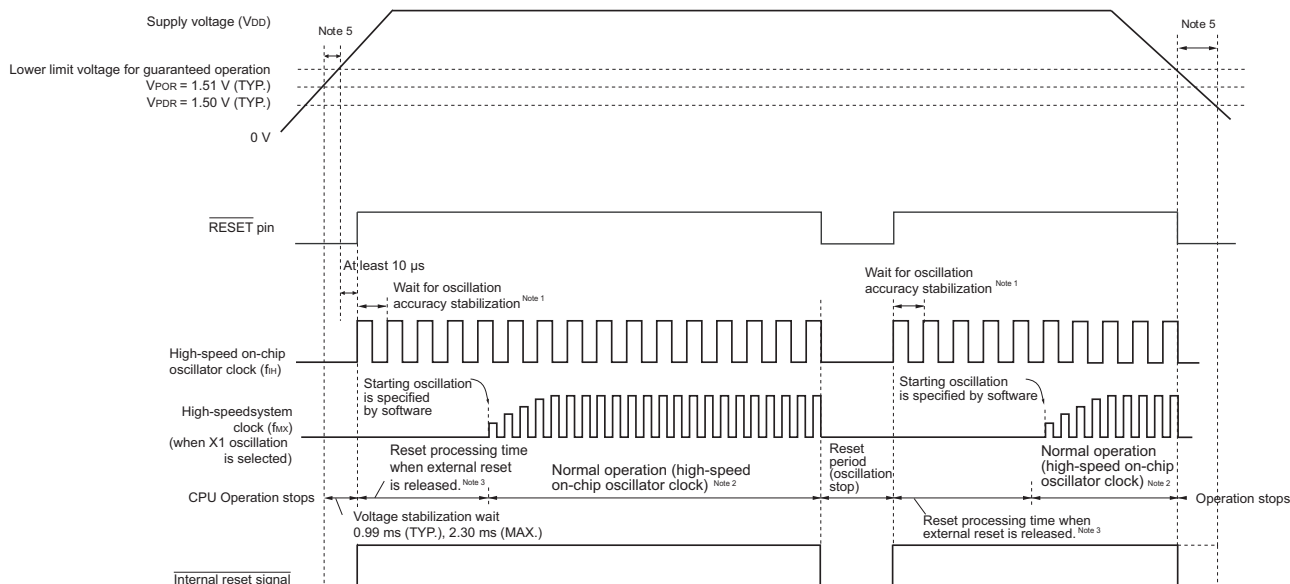


### 23.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown below.

Figure 23-2. 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.

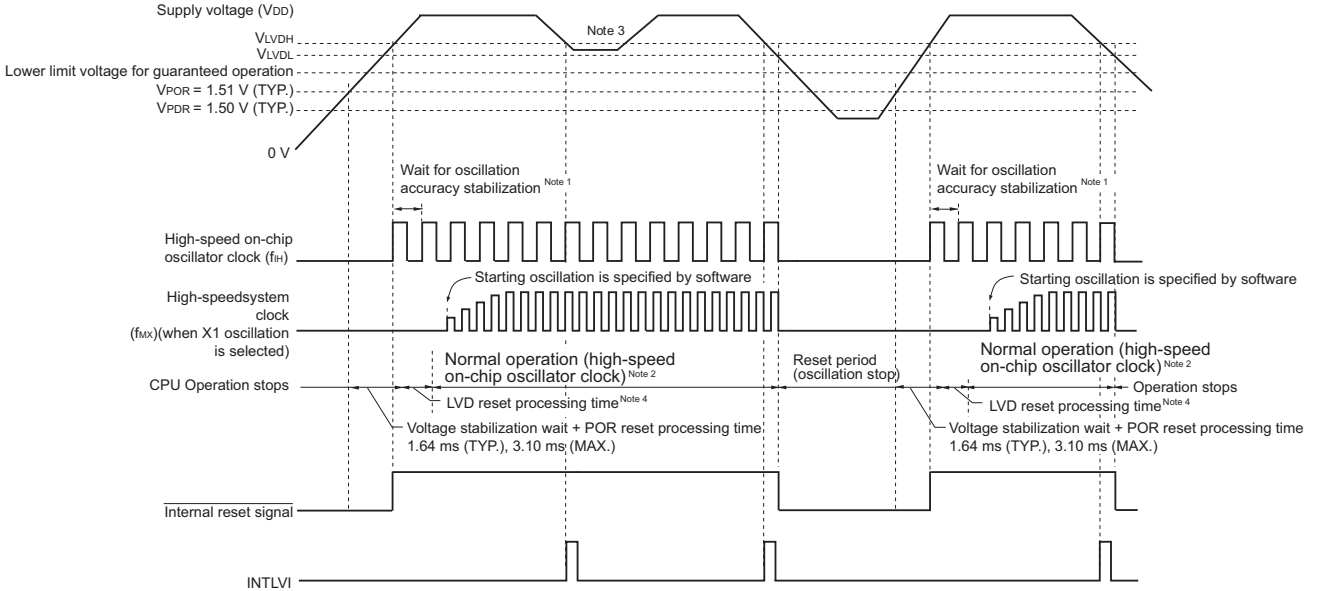
- <R>
3. The time until normal operation starts includes the following reset processing time when the external reset is released (release from the first external reset following release from the POR state) after the  $\overline{\text{RESET}}$  signal is driven high (1) as well as the voltage stabilization wait time after VPOR (1.51 V, typ.) is reached. Reset processing time when the external reset is released is shown below.  
Release from the first external reset 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)
- <R>
4. Reset times in cases of release from an external reset other than the above are listed below.  
Release from the reset state for external resets other than the above case:  
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 **32.4** or **33.4 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.

**Remark** VPOR: POR power supply rise detection voltage  
VPDR: POR power supply fall detection voltage

**Caution** For power-on reset, be sure to use the externally input reset signal on the  $\overline{\text{RESET}}$  pin when the LVD is off. For details, see CHAPTER 24 VOLTAGE DETECTOR.

**Figure 23-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)**

**(2) When LVD is in interrupt & reset mode (option byte 000C1H: LVIMDS1, LVIMDS0 = 1, 0)**



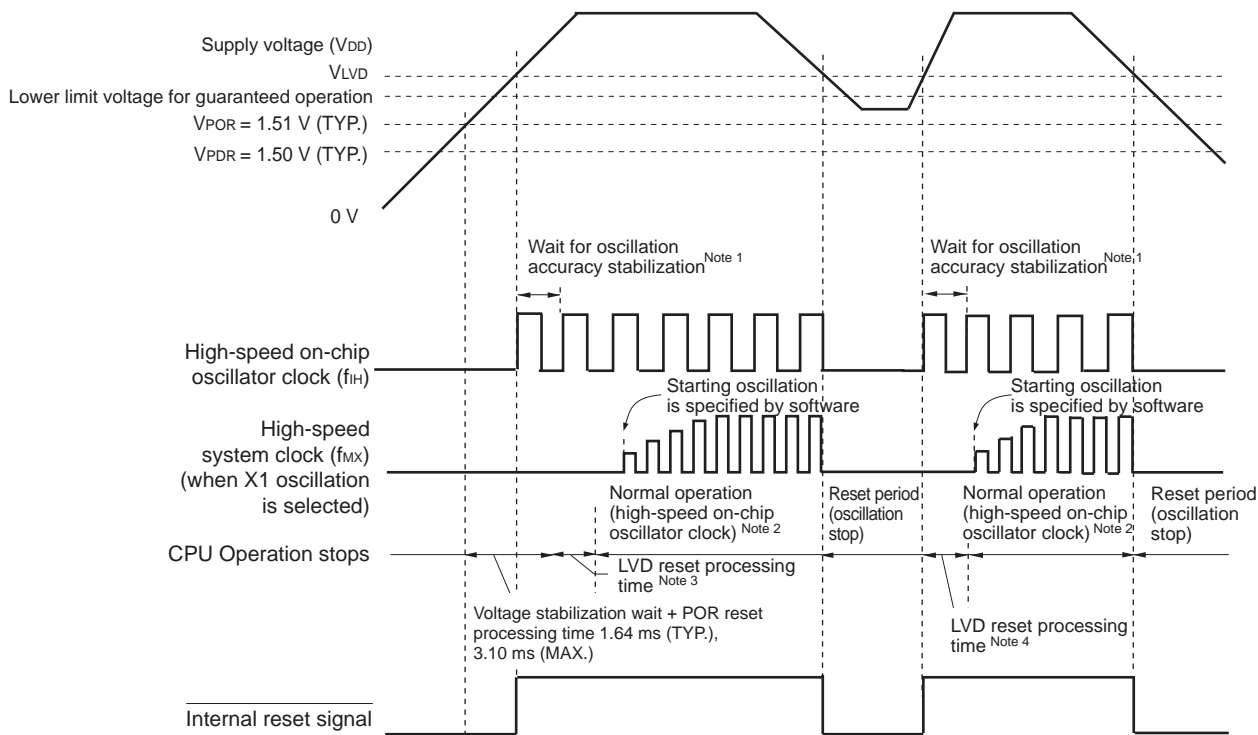
- Notes**
1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.
  2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the 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 24 - 8 Setting Procedure for Operating Voltage Check and Reset** and **Figure 24 - 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 (VLVDH) or higher without falling below the low-voltage detection level (VLVDL).
  4. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (VLVDH) is reached as well as the voltage stabilization wait + POR reset processing time after the VPOr (1.51 V, typ.) is reached.  
LVD reset processing time: 0 ms to 0.0701 ms (max.)

<R>

**Remark** VLVDH, VLVDL: LVD detection voltage  
 VPOr: POR power supply rise detection voltage  
 VPDR: POR power supply fall detection voltage

**Figure 23-2. 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<sub>LVD</sub>) is reached as well as the voltage stabilization wait + POR reset processing time after the V<sub>POR</sub> (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<sub>LVD</sub>) is reached.  
LVD reset processing time: 0.0511 ms (typ.), 0.0701 ms (max.)

- Remarks**
1. V<sub>LVDH</sub>, V<sub>LVDL</sub>: LVD detection voltage  
V<sub>POR</sub>: POR power supply rise detection voltage  
V<sub>PDR</sub>: POR power supply fall detection voltage
  2. When the LVD interrupt mode is selected (option byte 000C1H: LVIMD1 = 0, LVIMD0 = 1), the time until normal operation starts after power is turned on is the same as the time specified in Note 3 of Figure 23-2 (3).

CHAPTER 24 VOLTAGE DETECTOR

24.1 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 14 levels (For details, see **CHAPTER 27 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 **32.4** or **34.4 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 detection circuit 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 generating/releasing resets.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)

The detection voltage ( $V_{LVD}$ ) selected by the option byte 000C1H is used for releasing resets/generating interrupts.

The interrupt signals and internal reset 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 interrupt request signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage falls, and releases an internal reset by detecting $V_{DD} < V_{LVDH}$ . Releases an internal reset by detecting $V_{DD} \geq V_{LVDH}$ .	Releases an internal reset by detecting $V_{DD} \geq V_{LVDL}$ . Generates an internal reset by detecting $V_{DD} < V_{LVDL}$ .	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.

<R>

While the voltage detector is operating, whether the supply voltage or the input voltage from an external input pin 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)).

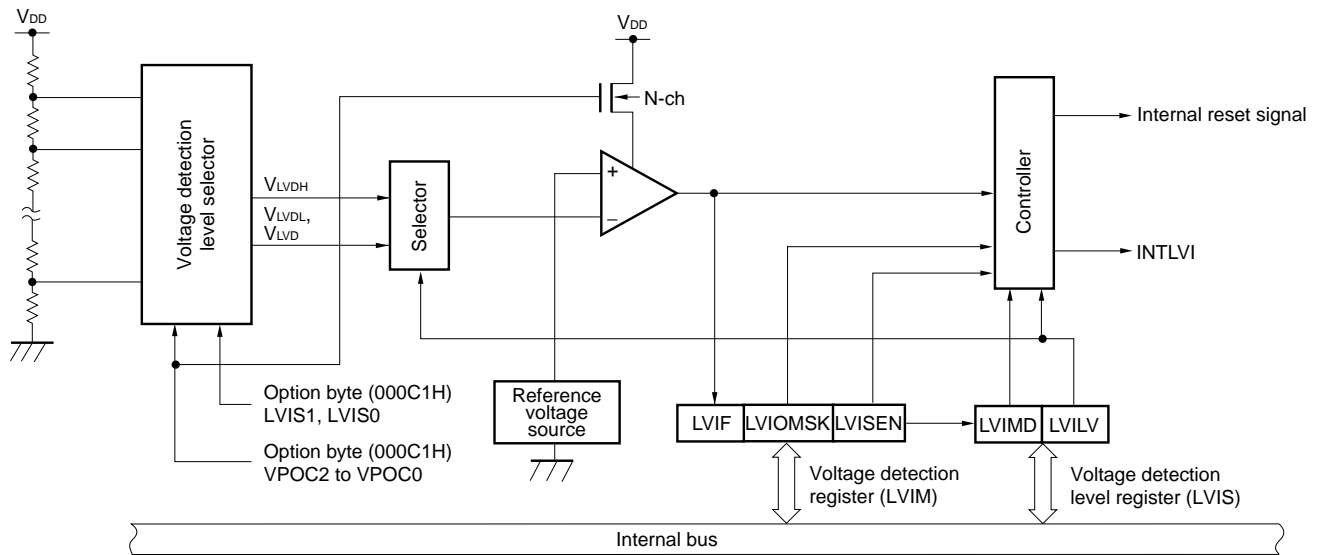
Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 22 RESET FUNCTION**.



### 24.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 24-1.

Figure 24-1. Block Diagram of Voltage Detector



### 24.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

**24.3.1 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 24-2. Format of Voltage Detection Register (LVIM)**

Address: FFFA9H After reset: 00H<sup>Note 1</sup> R/W<sup>Note 2</sup>

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVISEN <sup>Note 3</sup>	0	0	0	0	0	LVIOMSK	LVIF

LVISEN <sup>Note 3</sup>	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 <sup>Note 4</sup>

LVIF	Voltage detection flag
0	Supply voltage (V <sub>DD</sub> ) ≥ detection voltage (V <sub>LVD</sub> ), or when LVD is off
1	Supply voltage (V <sub>DD</sub> ) < detection voltage (V <sub>LVD</sub> )

- 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. This can be set only in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not change the initial value in other modes.
  4. LVIOMSK bit is automatically set to “1” only in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0) and reset or interrupt 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

**24.3.2 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 sets this register to 00H/01H/81H<sup>Note 1</sup>.

**Figure 24-3. Format of Voltage Detection Level Select Register (LVIS)**

Address: FFFAAH    After reset: 00H/01H/81H<sup>Note 1</sup>    R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD	0	0	0	0	0	0	LVILV

LVIMD <sup>Note 2</sup>	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVILV <sup>Note 2</sup>	LVD detection level
0	High-voltage detection level (VLVDH)
1	Low-voltage detection level (VLVDL or VLVD)

- Notes**
- 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
  - 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**
- Rewrite the value of the LVIS register according to Figures 24-8.
  - Specify the LVD operation mode and detection voltage (VLVDH, VLVDL, VLVD) of each mode by using the option byte 000C1H. Figure 24-4 shows the format of the user option byte (000C1H/010C1H). For details about the option byte, see CHAPTER 27 OPTION BYTE.

**Figure 24-4. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H) (1/2)**

Address: 000C1H/010C1H<sup>Note</sup>

	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						
VLVDH		VLVDL	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
1.77 V	1.73 V	1.63 V	0	0	0	1	0	1	0
1.88 V	1.84 V					0	1		
2.92 V	2.86 V					0	0		
1.98 V	1.94 V	1.84 V		0	1	1	0		
2.09 V	2.04 V					0	1		
3.13 V	3.06 V					0	0		
2.61 V	2.55 V	2.45 V		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						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.67 V	1.63 V	0	0	0	1	1	1	1
1.77 V	1.73 V		0	0	1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	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.

**Remark** The detection voltage is a TYP. value. For details, see **32.6.5** or **33.6.5 LVD circuit characteristics**.

**Figure 24-4. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H) (2/2)**

Address: 000C1H/010C1H<sup>Note</sup>

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 <sub>LVD</sub>		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting				
Rising edge	Falling edge						LVIMDS1	LVIMDS0			
1.67 V	1.63 V	0	0	0	1	1	0	1			
1.77 V	1.73 V		0	0	1	0					
1.88 V	1.84 V		0	1	1	1					
1.98 V	1.94 V		0	1	1	0					
2.09 V	2.04 V		0	1	0	1					
2.50 V	2.45 V		1	0	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.								

• LVD off (use of external reset input via  $\overline{\text{RESET}}$  pin)

Detection voltage		Option byte setting value						
V <sub>LVD</sub>		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 32.4 or 33.4 AC Characteristics. This is done by utilizing the voltage detection circuit 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 detection circuit 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 32.6.5 or 33.6.5 LVD circuit characteristics.

## 24.4 Operation of Voltage Detector

### 24.4.1 When used as reset mode

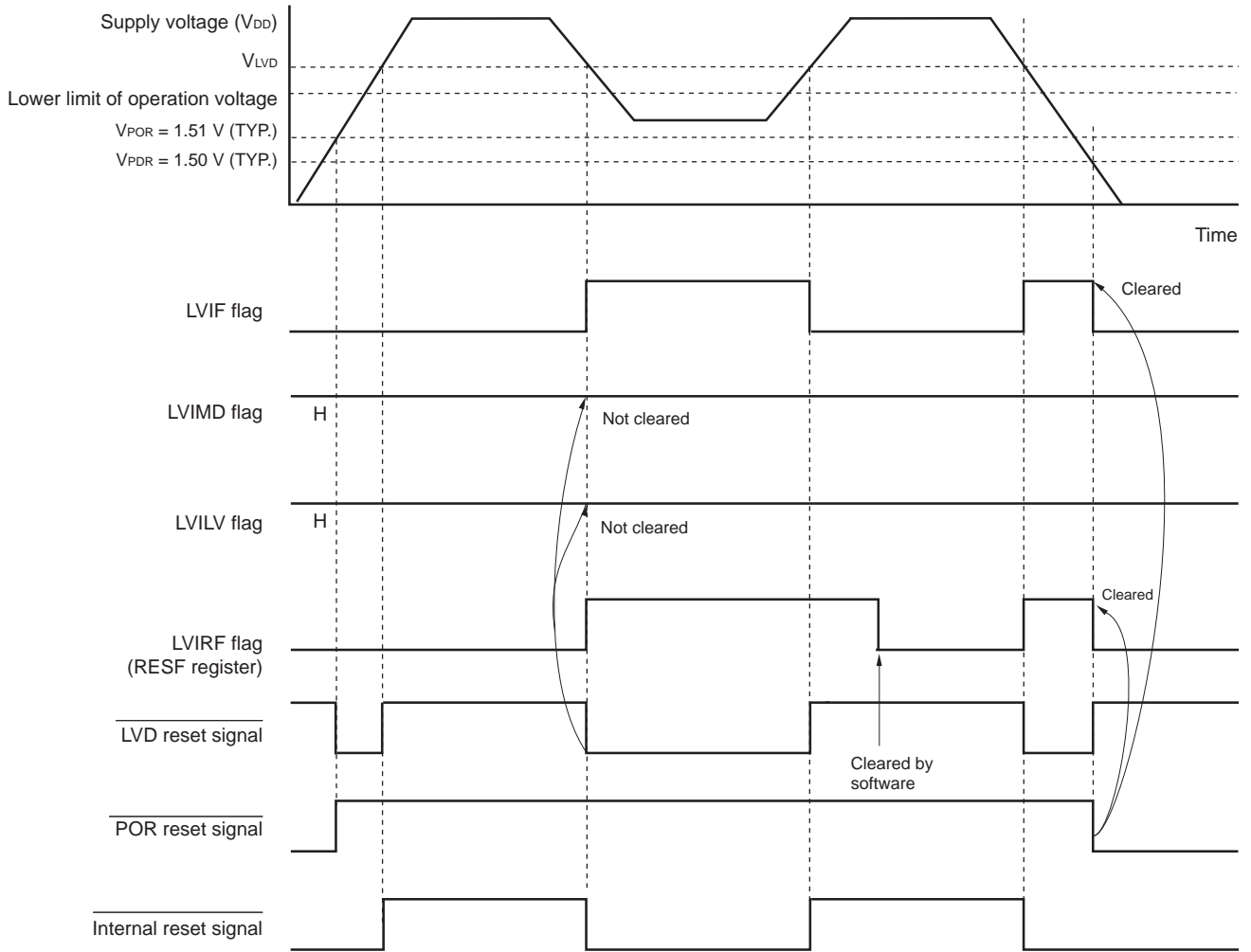
Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the detection voltage ( $V_{LVD}$ ) by using the option byte 000C1H.

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 retained until the supply voltage ( $V_{DD}$ ) exceeds the voltage detection level ( $V_{LVD}$ ) after power is supplied. The internal reset is released when the supply voltage ( $V_{DD}$ ) exceeds the voltage detection level ( $V_{LVD}$ ).
  - At the fall of the operating voltage, an internal reset by LVD is generated when the supply voltage ( $V_{DD}$ ) falls below the voltage detection level ( $V_{LVD}$ )

Figure 24-5 shows the timing of the internal reset signal generated in the LVD reset mode.

**Figure 24-5. 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

### 24.4.2 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.

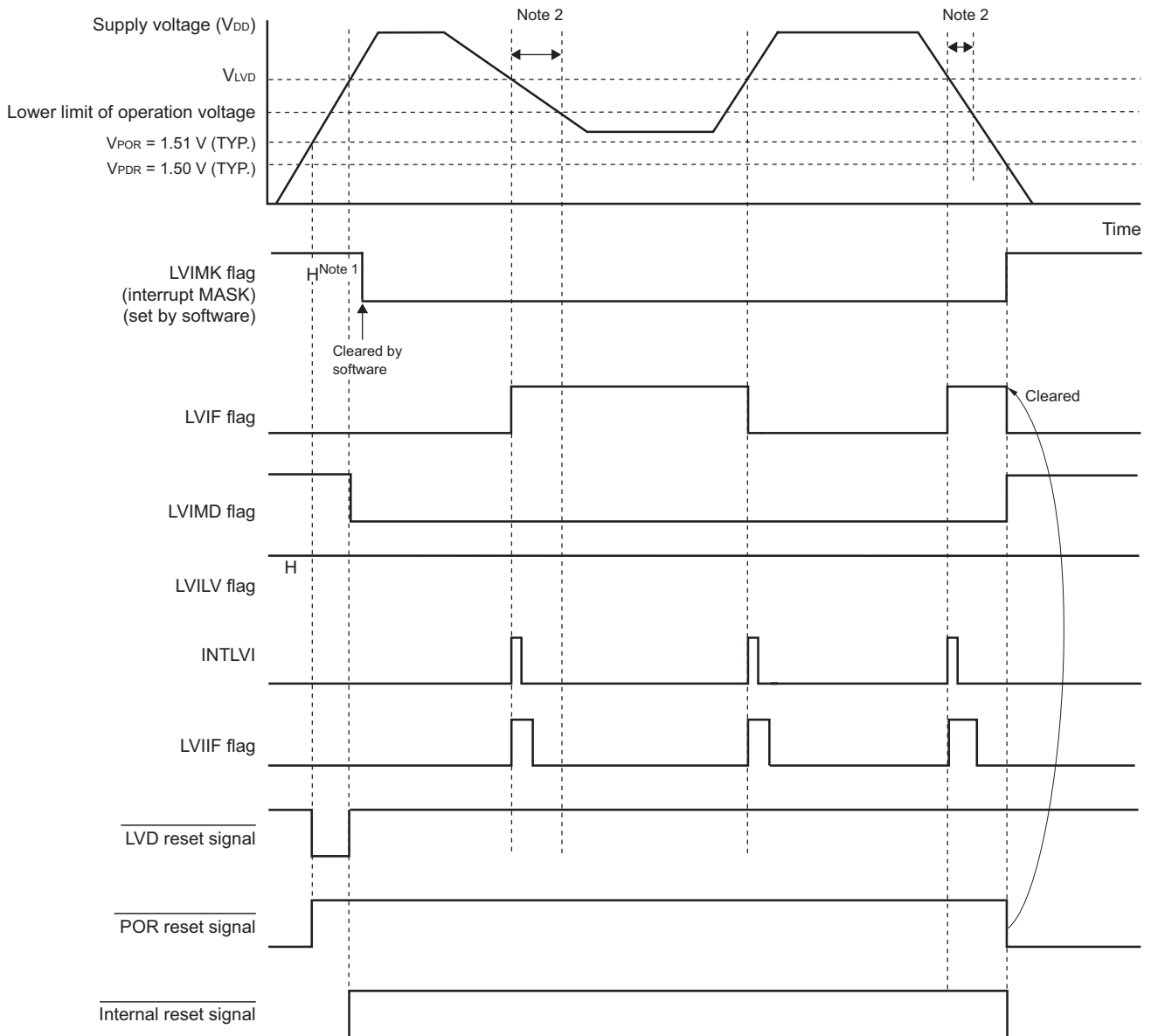
The operation is started in the following initial setting state when the interrupt mode is set.

- 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
  - The state of an internal reset by LVD is retained until  $V_{DD} \geq V_{LVD}$  immediately after reset generation. The internal reset is released when  $V_{DD} \geq V_{LVD}$  is detected.
  - After that, an interrupt request signal (INTLVI) is generated when  $V_{DD} < V_{LVD}$  or  $V_{DD} \geq V_{LVD}$  is detected.

Figure 24-6 shows the timing of the interrupt request signal generated in the LVD interrupt mode.



**Figure 24-6. 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. 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 32.4 or 33.4 AC Characteristics. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

**Remark**  $V_{POR}$ : POR power supply rise detection voltage  
 $V_{PDR}$ : POR power supply fall detection voltage

### 24.4.3 When used as interrupt and reset mode

Specify the operation mode (the interrupt and 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 retained until the supply voltage ( $V_{DD}$ ) exceeds the high-voltage detection level ( $V_{LVDH}$ ) after power is supplied.

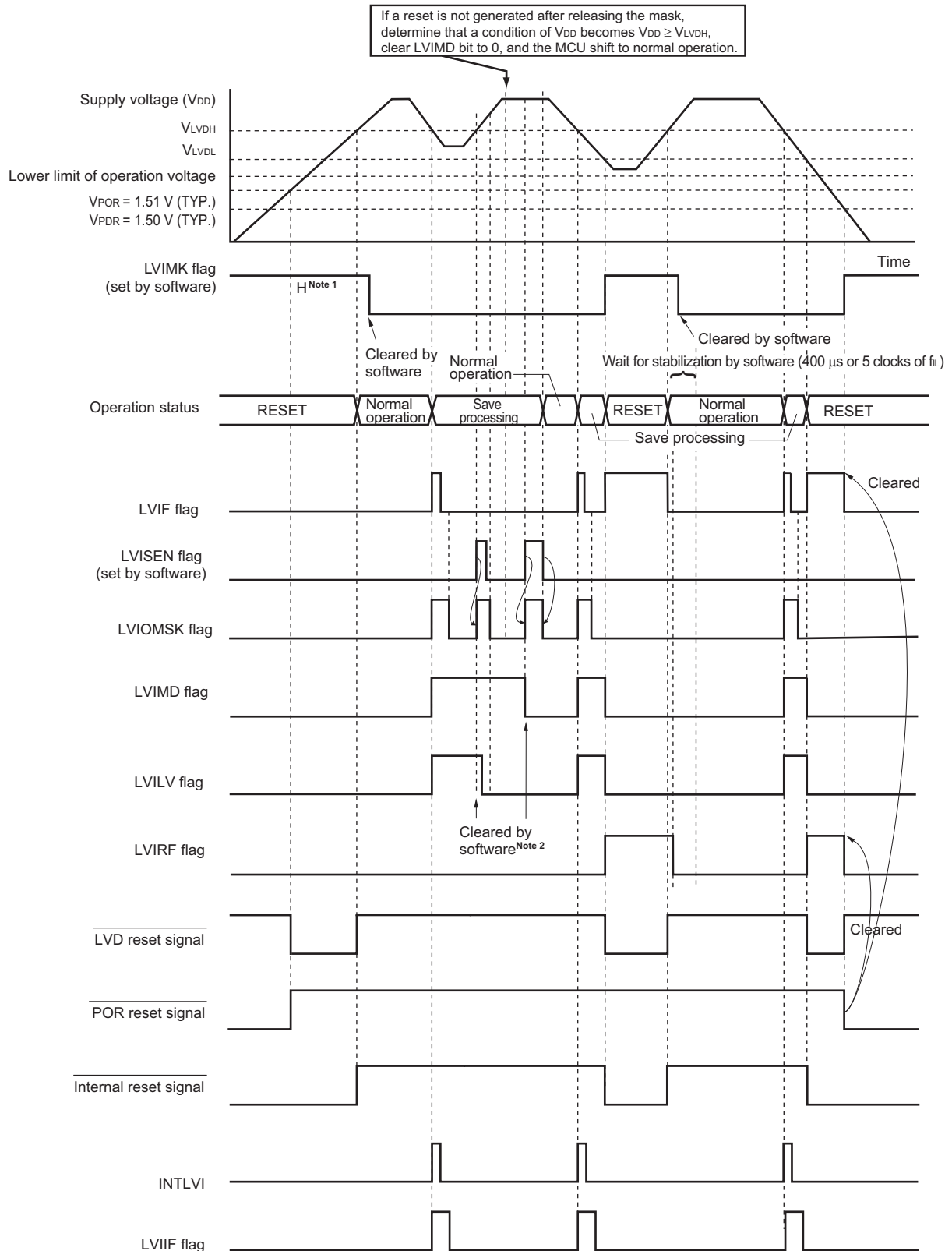
The internal reset is released when the supply voltage ( $V_{DD}$ ) exceeds the high-voltage detection level ( $V_{LVDH}$ ).

An interrupt request signal by LVD (INTLVI) is generated and arbitrary save processing is performed when the supply voltage ( $V_{DD}$ ) falls below the high-voltage detection level ( $V_{LVDH}$ ). After that, an internal reset by LVD is generated when the supply voltage ( $V_{DD}$ ) falls below the low-voltage detection level ( $V_{LVDL}$ ). After INTLVI is generated, an interrupt request signal is not generated even if the supply voltage becomes equal to or higher than the high-voltage detection voltage ( $V_{LVDH}$ ) without falling below the low-voltage detection voltage ( $V_{LVDL}$ ).

To use the LVD reset & interrupt mode, perform the processing according to **Figure 24-8 Setting Procedure for Operating Voltage Check/Reset**.

Figure 24-7 shows the timing of the internal reset signal and interrupt signal generated in the LVD interrupt & reset mode.

**Figure 24-7. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)**

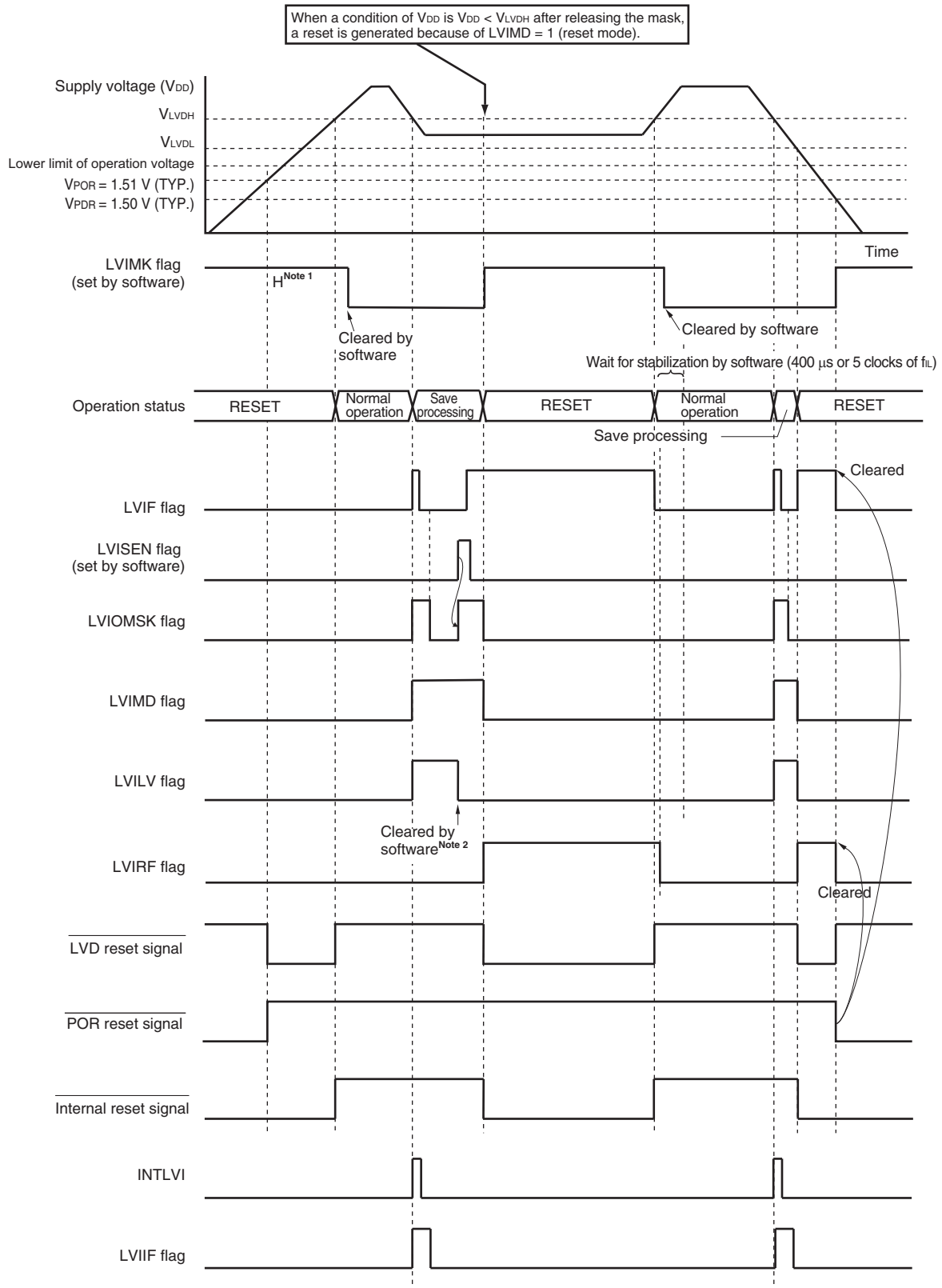


(Notes and Remark are listed on the next page.)

- 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 24-8 Setting Procedure for Operating Voltage Check/Reset** in interrupt and reset mode.

**Remark**  $V_{POR}$ : POR power supply rise detection voltage  
 $V_{PDR}$ : POR power supply fall detection voltage

**Figure 24-7. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)**

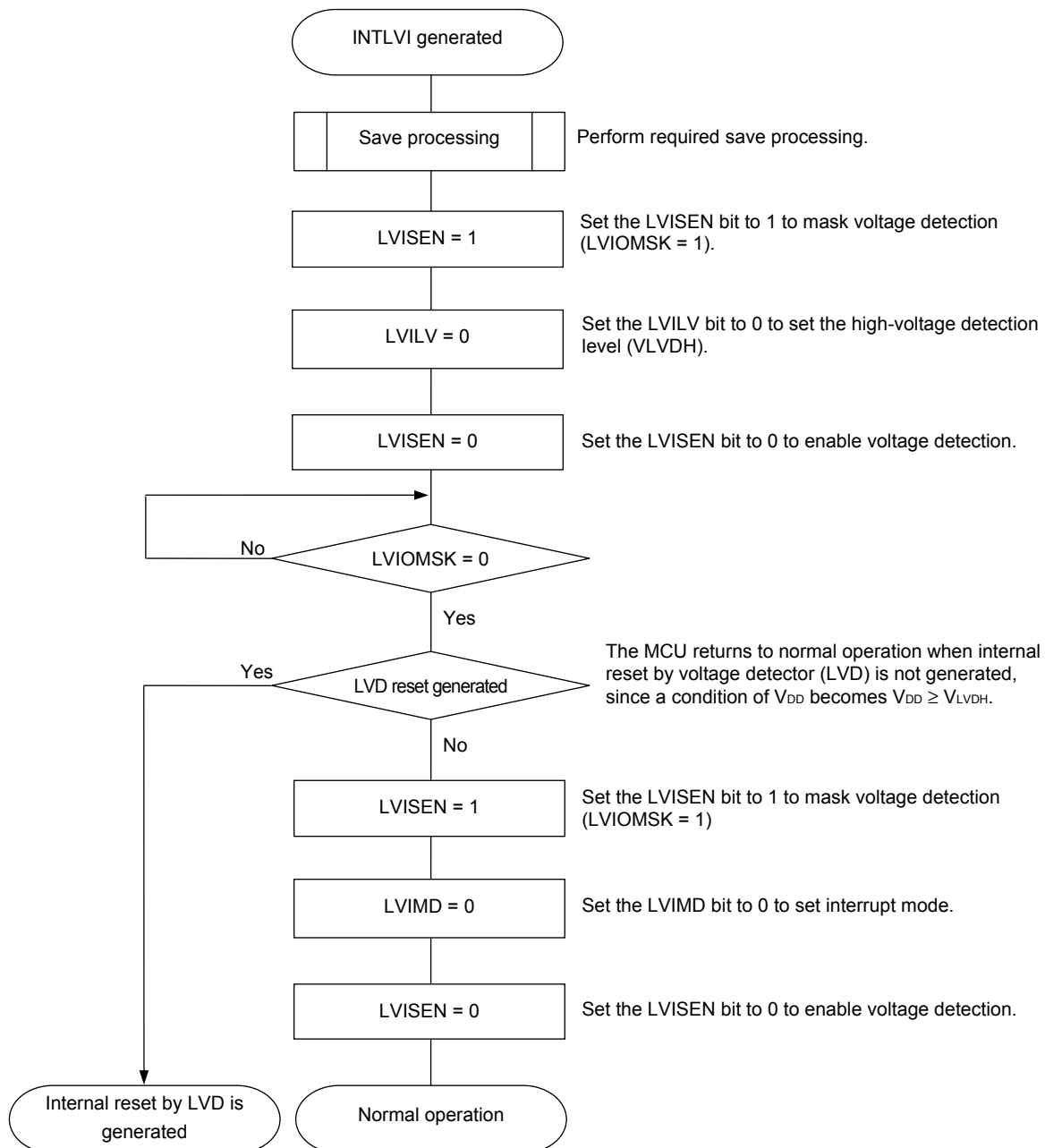


(Notes and Remark are listed on the next page.)

- 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 24-8 Setting Procedure for Operating Voltage Check/Reset** in interrupt and reset mode.

**Remark**  $V_{POR}$ : POR power supply rise detection voltage  
 $V_{PDR}$ : POR power supply fall detection voltage

**Figure 24-8. Setting Procedure for Operating Voltage Check/Reset**



### 24.5 Cautions for Voltage Detector

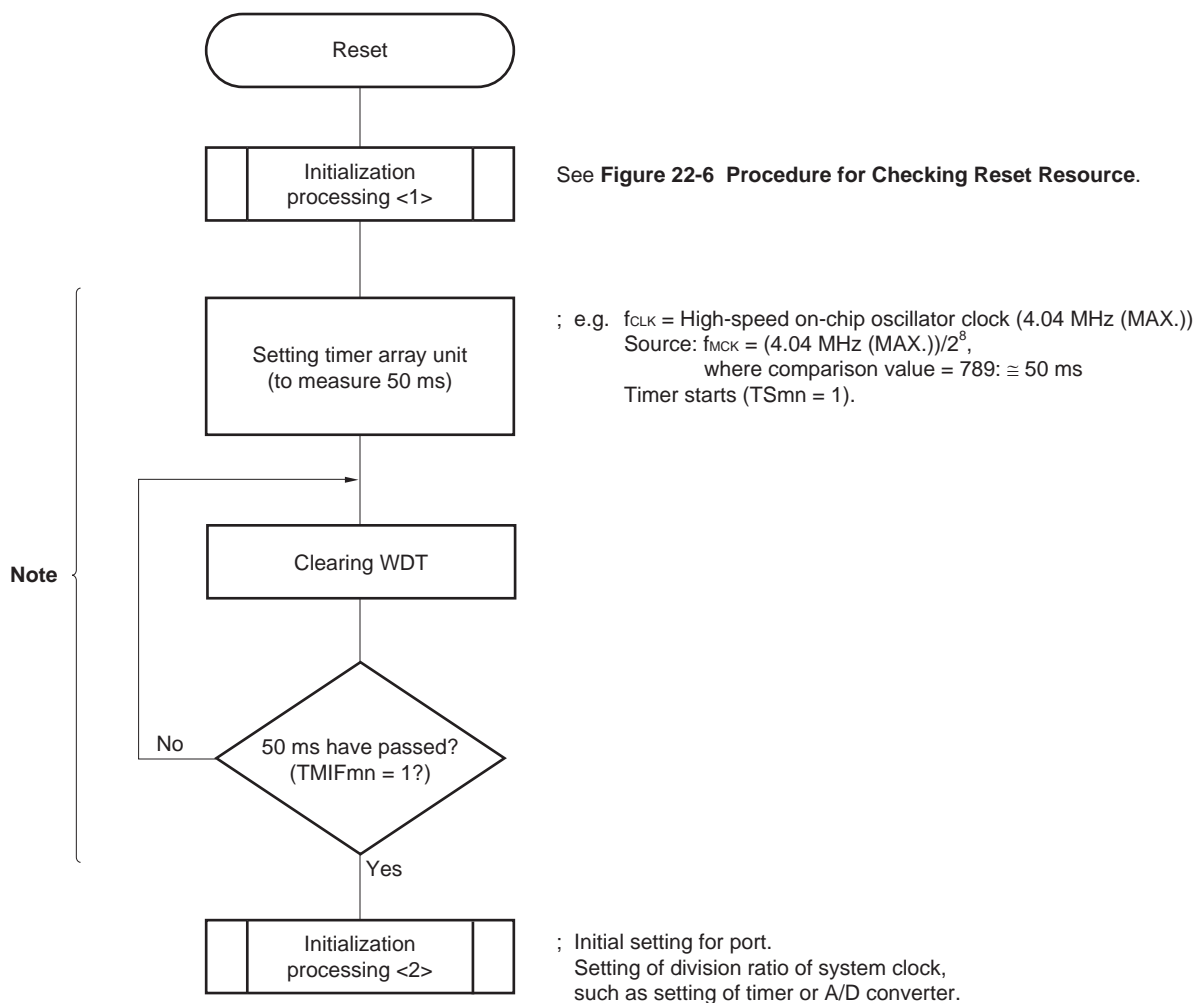
#### (1) Voltage fluctuation when power is supplied

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 24-9. 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.

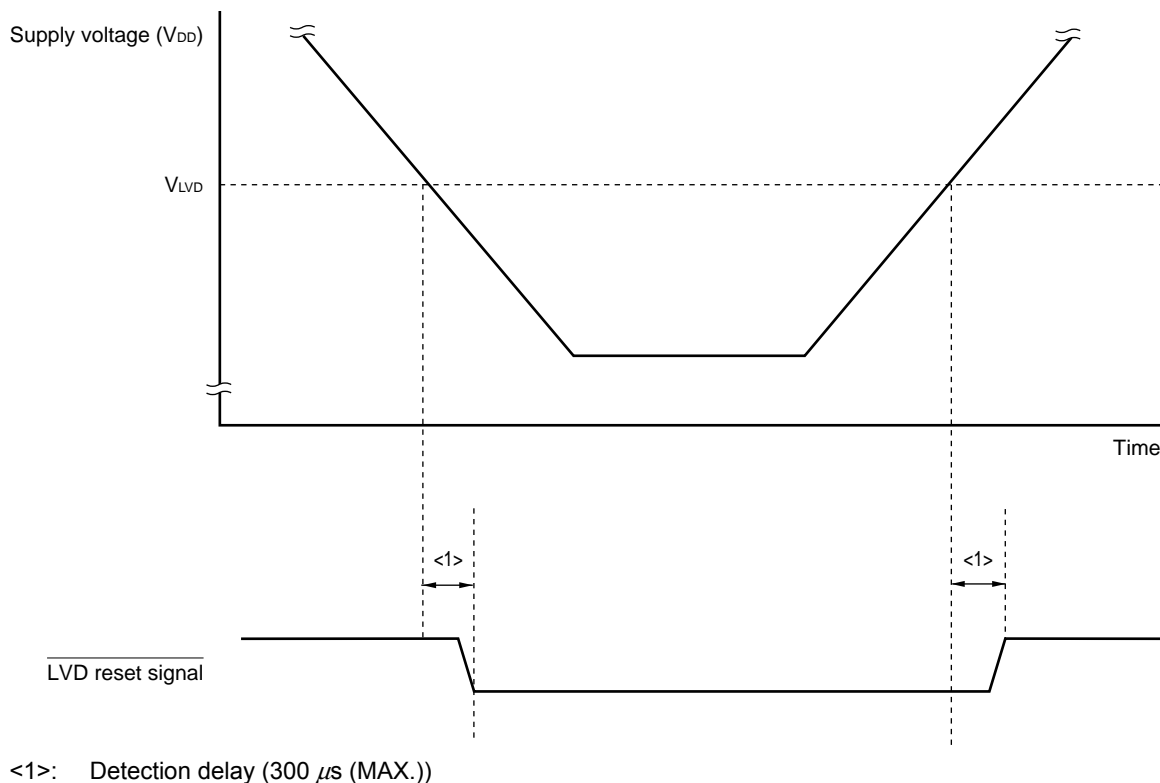
**Remark** m = 0  
 n = 0 to 7

**(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 ( $V_{LVD}$ )  $\leq$  supply voltage ( $V_{DD}$ ) until the time LVD reset has been released (see **Figure 24-10**).

**Figure 24-10. Delay from the Time LVD Reset Source Is Generated Until the Time LVD Reset has Been Generated or Released**

**(3) Power on when LVD is off**

Use the external reset input via the  $\overline{\text{RESET}}$  pin when the LVD is off.

For an external reset, input a low level for 10  $\mu$ s 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  $\mu$ s or more within the operating voltage range shown in **32.4** or **33.4 AC Characteristics**, and then input a high level to the pin.

**(4) Operating voltage fall when LVD is off or LVD interrupt mode is selected**

When the operating voltage falls with the LVD is off or with the LVD interrupt mode is selected, 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 **32.4** or **33.4 AC characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.



## CHAPTER 25 SAFETY FUNCTIONS

### 25.1 Overview of Safety Functions

The following safety functions are provided in the RL78/L13 to comply with the IEC60730 and IEC61508 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/L13 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 voltage, and internal reference voltage.

#### (8) Digital output signal level detection function for I/O pins

When the I/O pins are output mode, the output level of the pin can be read.

**Remark** For usage examples of the safety functions complying with the IEC60730 safety standards, refer to the **RL78 MCU series IEC60730/60335 Application Notes (R01AN0749)**.

## 25.2 Registers Used by Safety Functions

The safety functions use the following registers for each function.

Register	Each Function of Safety Function
<ul style="list-style-type: none"> <li>Flash memory CRC control register (CRC0CTL)</li> <li>Flash memory CRC operation result register (PGCRCL)</li> </ul>	Flash memory CRC operation function (high-speed CRC)
<ul style="list-style-type: none"> <li>CRC input register (CRCIN)</li> <li>CRC data register (CRCD)</li> </ul>	CRC operation function (general-purpose CRC)
<ul style="list-style-type: none"> <li>RAM parity error control register (RPECTL)</li> </ul>	RAM parity error detection function
<ul style="list-style-type: none"> <li>Invalid memory access detection control register (IAWCTL)</li> </ul>	RAM guard function
	SFR guard function
	Invalid memory access detection function
<ul style="list-style-type: none"> <li>Timer input select register 0 (TIS0)</li> </ul>	Frequency detection function
<ul style="list-style-type: none"> <li>A/D test register (ADTES)</li> </ul>	A/D test function
<ul style="list-style-type: none"> <li>Port mode select register (PMS)</li> </ul>	Digital output signal level detection function for I/O pins

The content of each register is described in **25.3 Operation of Safety Functions**.

## 25.3 Operation of Safety Functions

### 25.3.1 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/L13 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, 341  $\mu$ s@24 MHz with 32-KB flash memory). 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.

**25.3.1.1 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 25-1. 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	0	0	0	FEA2 <sup>Note</sup>	FEA1	FEA0

CRC0EN	Control of CRC ALU operation
0	Stop the operation.
1	Start the operation according to HALT instruction execution.

FEA2 <sup>Note</sup>	FEA1	FEA0	High-speed CRC operation range
0	0	0	0000H to 3FFBH (16 K – 4 bytes)
0	0	1	0000H to 7FFBH (32 K – 4 bytes)
0	1	0	0000H to BFFBH (48 K – 4 bytes)
0	1	1	0000H to FFFBH (64 K – 4 bytes)
1	0	0	00000H to 13FFBH (80 K – 4 bytes)
1	0	1	00000H to 17FFBH (96 K – 4 bytes)
1	1	0	00000H to 1BFFBH (112 K – 4 bytes)
1	1	1	00000H to 1FFFBH (128 K – 4 bytes)

**Note** Be sure to clear bit 2 to “0” in products with a 64 KB flash memory.

**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.

### 25.3.1.2 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 25-2. Format of Flash Memory CRC Operation Result Register (PGCRCL)**

Address: F02F2H After reset: 0000H R/W

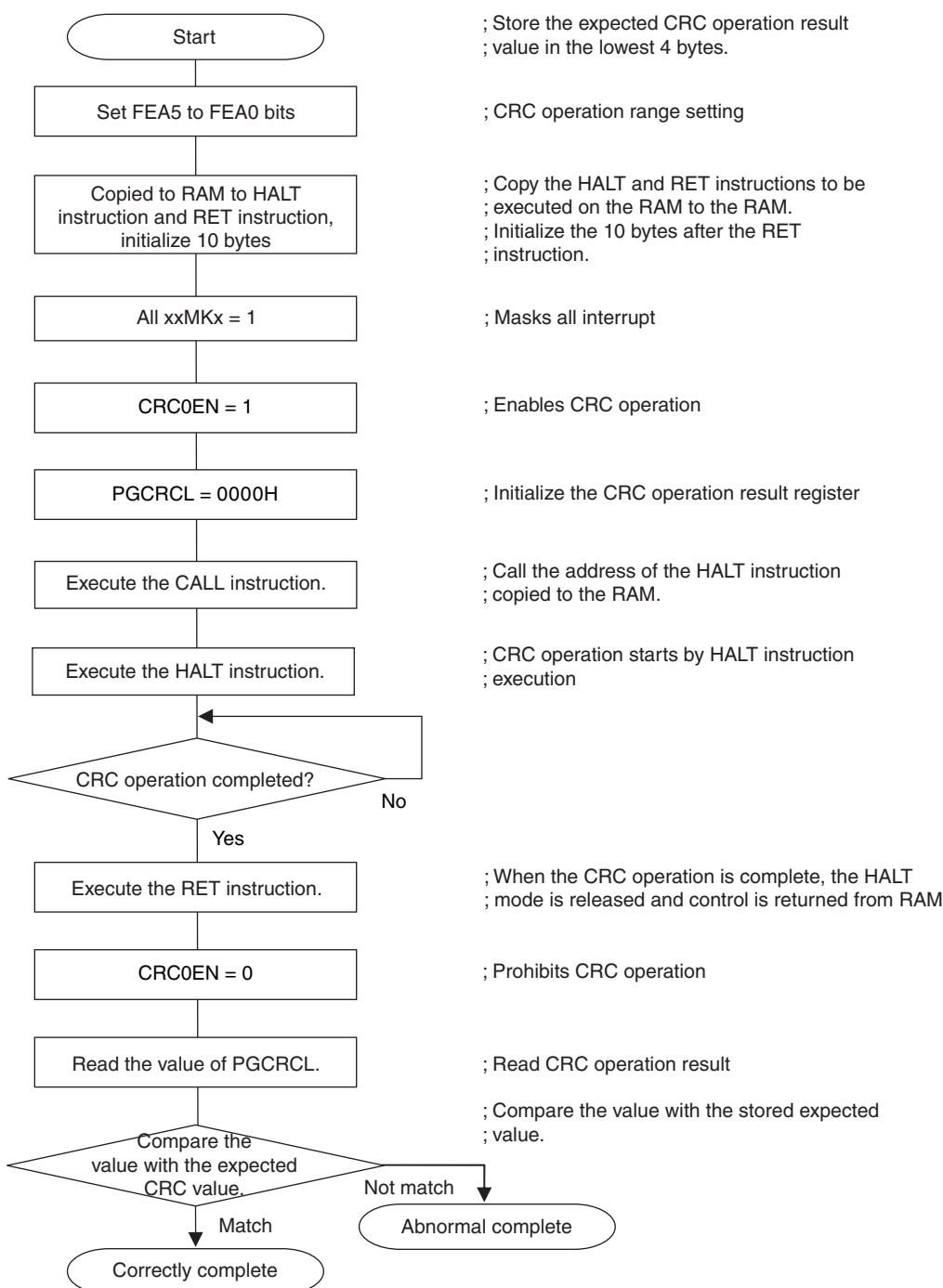
Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8
	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0
	PGCRC15 to 0		High-speed CRC operation results					
	0000H to FFFFH		Store the high-speed CRC operation results.					

**Caution** The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 25-3 shows the flowchart of flash memory CRC operation function (high-speed CRC).

<Operation flow>

Figure 25-3. 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.

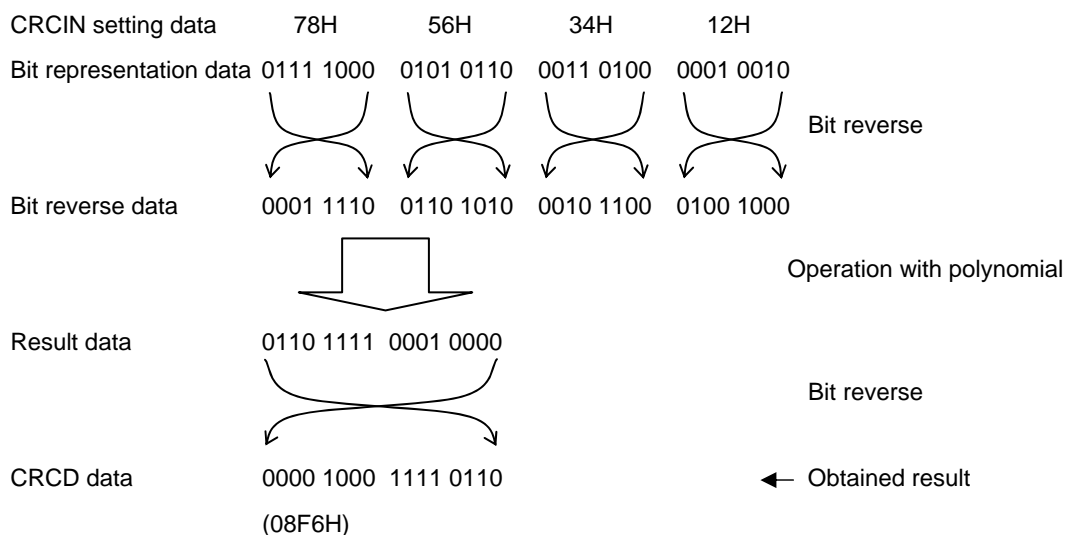
**25.3.2 CRC operation function (general-purpose CRC)**

In order to guarantee safety during operation, the IEC61508 standard mandates the checking of data even while the CPU is operating.

In the RL78/L13, 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.

**25.3.2.1 CRC input register (CRCIN)**

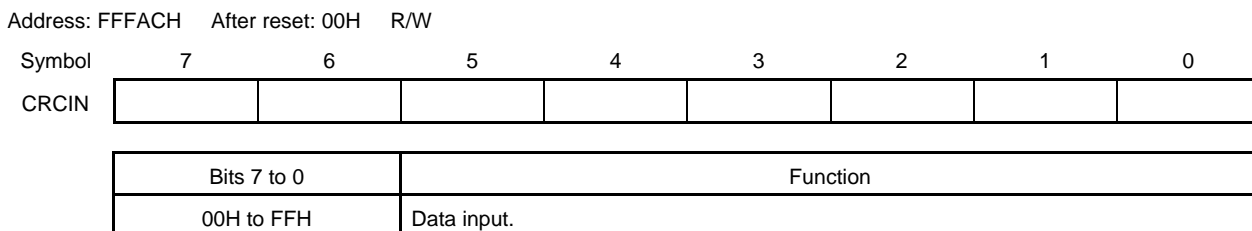
CRCIN register is an 8-bit register that is used to set the CRC operation data of general-purpose CRC.

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 25-4. Format of CRC Input Register (CRCIN)**



**25.3.2.2 CRC data register (CRCD)**

This register is used to store the CRC operation result of the general-purpose CRC.

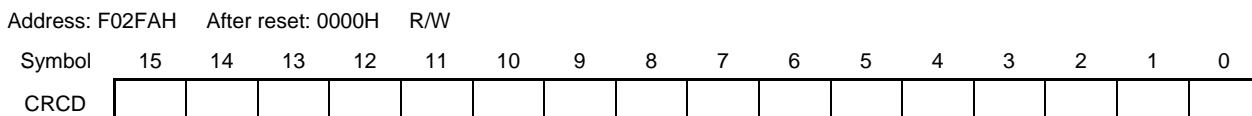
The setting range is 0000H to FFFFH.

When one cycle of the CPU/peripheral hardware clock ( $f_{CLK}$ ) has elapsed after 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.

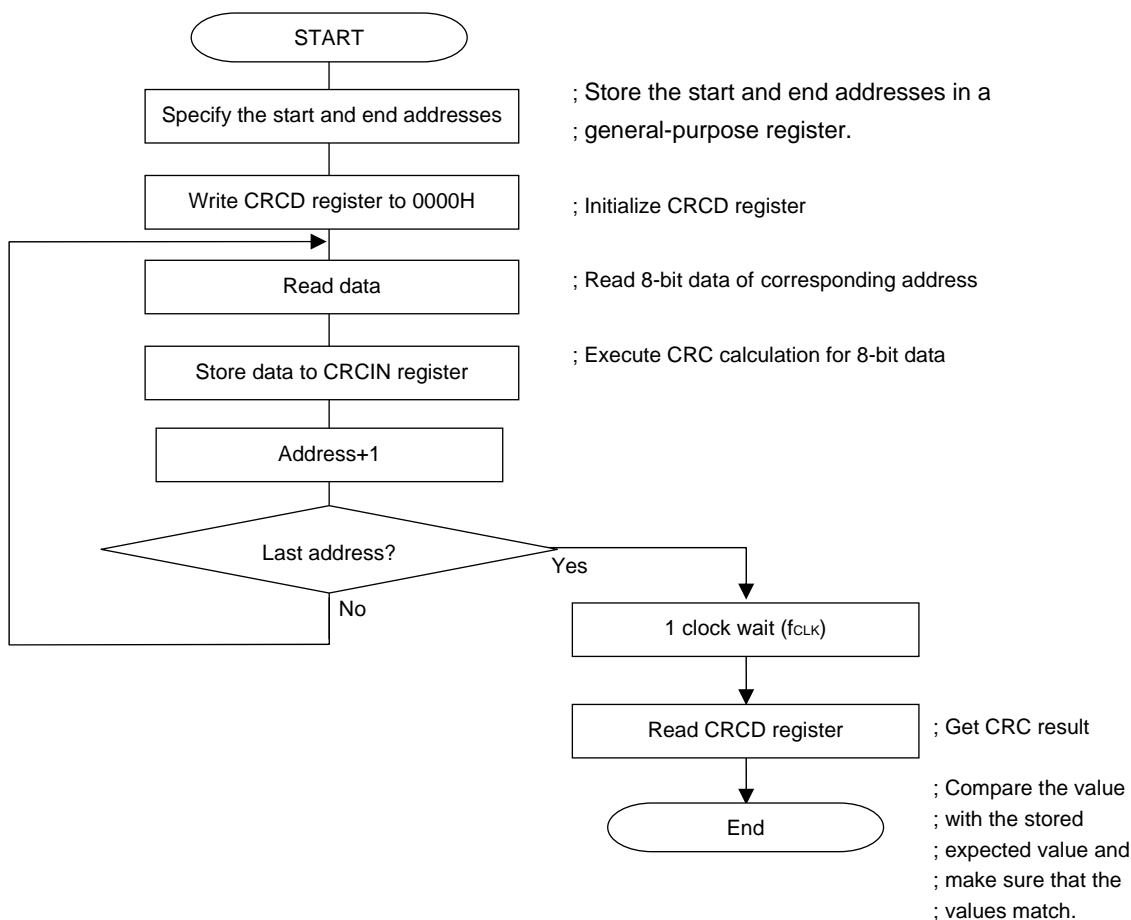
**Figure 25-5. Format of CRC Data Register (CRCD)**



- 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 25-6. CRC Operation Function (General-Purpose CRC)**



### 25.3.3 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/L13'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.

#### 25.3.3.1 RAM parity error control register (RPECTL)

This register is used to control the 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 25-7. 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.

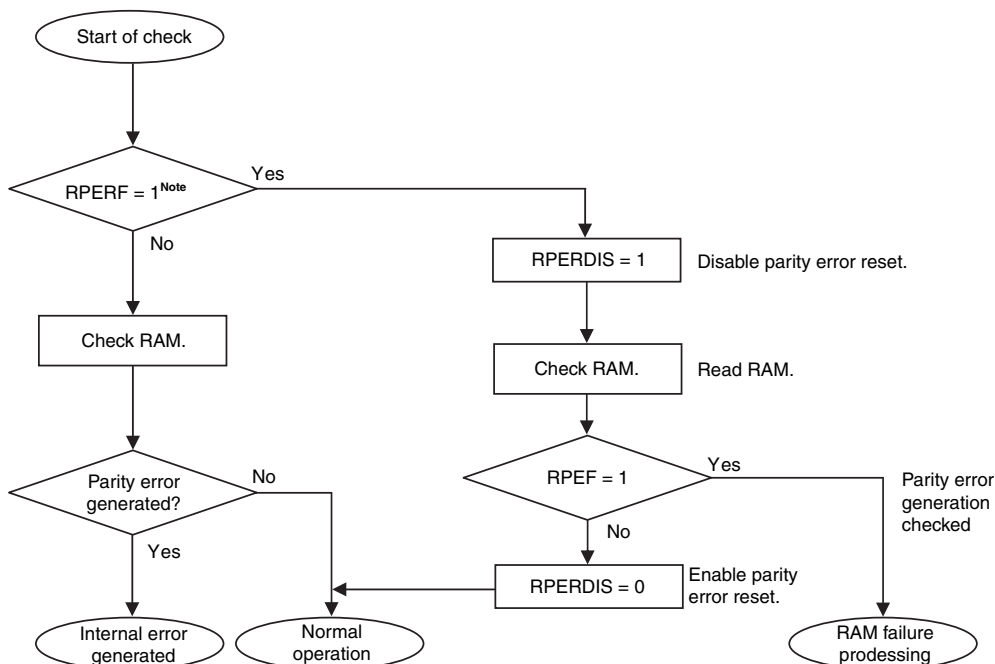
RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

**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 RPECTL register is set (1) when the RAM parity error occurs 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 25-8. Flowchart of RAM Parity Check



**Note** To check internal reset status using a RAM parity error, see **CHAPTER 22 RESET FUNCTION**.

**25.3.4 RAM guard function**

In order to guarantee safety during operation, the IEC61508 standard mandates that important data stored in the RAM be protected, even if the CPU freezes.

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.

**25.3.4.1 Invalid memory access detection control register (IAWCTL)**

This register is used to control detection of invalid memory access and the RAM/SFR guard function.

The GRAM1 and GRAM0 bits are used for the 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 25-9. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H 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 <sup>Note</sup>
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes of space starting at the start address in the RAM
1	0	The 256 bytes of space starting at the start address in the RAM
1	1	The 512 bytes of space starting at the start address in the RAM

**Note** The RAM start address differs depending on the size of the RAM provided with the product.

### 25.3.5 SFR guard function

In order to guarantee safety during operation, the IEC61508 standard mandates that important SFRs be protected from being overwritten, even if the CPU freezes.

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.

#### 25.3.5.1 Invalid memory access detection control register (IAWCTL)

This register is used to control detection of invalid memory access and the RAM/SFR guard function.

The GPORT, GINT and GCSC bits are used for 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 25-10. Format of Invalid Memory Access Detection Control Register (IAWCTL)**

Address: F0078H 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, PFSEGxx, ISCLCD <sup>Note</sup>

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	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

**Note** Pxx (port register) is not guarded.

**25.3.6 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 25-11.

**Figure 25-11. Invalid Access Detection Area**

		Possibility access		
		Read	Write	Fetching instructions (execute)
FFFFH	Special function register (SFR) 256 byte			NG
FFF00H FFEFFH	General-purpose register 32 byte		OK	
FFEE0H FFEDFH				
	RAM <sup>Note</sup>			OK
zzzzzH	Mirror	OK	NG	NG
	Data flash memory		NG	NG
F1000H FOFFFH	Reserved			OK
F0800H F07FFH	Special function register (2nd SFR) 2 KB		OK	NG
F0000H EFFFFH	Reserved			OK
EF000H EEFFFH				NG
		NG	NG	NG
yyyyyH				
xxxxxH	Code flash memory <sup>Note</sup>			
00000H		OK		OK

**Note** The following table lists the code flash memory, RAM, and lowest detection address for each product:  
(Description is listed on the next page.)

Products	Code flash memory (00000H to xxxxxH)	RAM (zzzzzH to FFEFFH)	Detected lowest address for read/instruction fetch (execution) (yyyyyH)
R5F10WLA, R5F10WMA	16384 × 8 bits (00000H to 03FFFH)	1024 × 8 bits (FFB00H to FFEFFH)	10000H
R5F10WLC, R5F10WMC	32768 × 8 bits (00000H to 07FFFH)	1536 × 8 bits (FF900H to FFEFFH)	10000H
R5F10WLD, R5F10WMD	49152 × 8 bits (00000H to 0BFFFH)	2048 × 8 bits (FF700H to FFEFFH)	10000H
R5F10WLE, R5F10WME	65536 × 8 bits (00000H to 0FFFFH)	4096 × 8 bits (FEF00H to FFEFFH)	10000H
R5F10WLF, R5F10WMF	98304 × 8 bits (00000H to 17FFFH)	6144 × 8 bits (FE700H to FFEFFH)	20000H
R5F10WLG, R5F10WMG	131072 × 8 bits (00000H to 1FFFFH)	8192 × 8 bits (FDF00H to FFEFFH)	20000H

**25.3.6.1 Invalid memory access detection control register (IAWCTL)**

This register is used to control detection of invalid memory access and the RAM/SFR guard function.  
 The IAWEN bit is used for the 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 25-12. Format of Invalid Memory Access Detection Control Register (IAWCTL)**

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN <sup>Note</sup>	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 valid; not writing 0 to the IAWEN bit is ignored after it is set to 1.

**Remark** By specifying WDTON = 1 (watchdog timer operation enable) for the option byte (000C0H), the invalid memory access function is enabled even IAWEN = 0.

### 25.3.7 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 5 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>

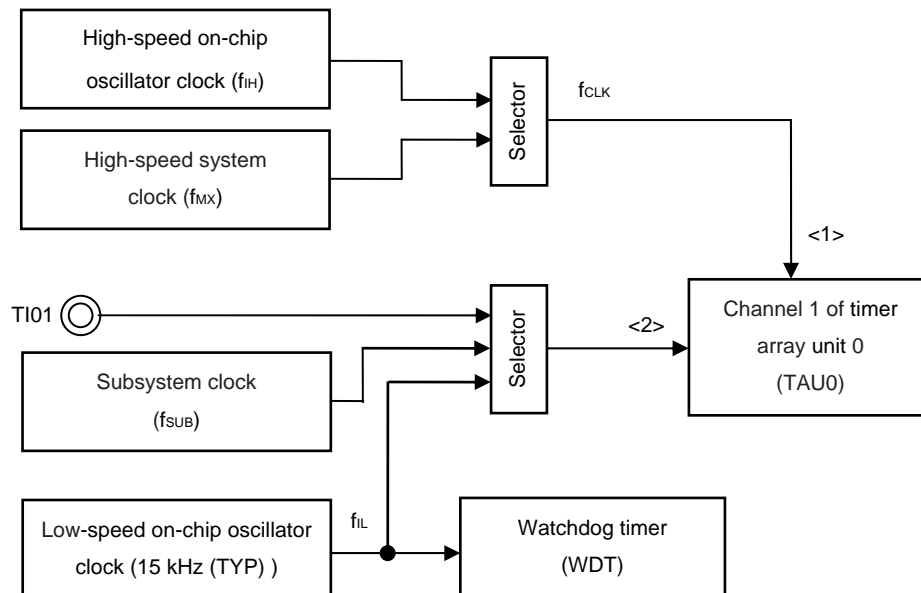
<1> CPU/peripheral hardware clock frequency ( $f_{CLK}$ ):

- 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 ( $f_{SUB}$ )

Figure 25-13. 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 **6.8.3 Operation as input pulse interval measurement**.

### 25.3.7.1 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input of channel 1 of 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 25-14. 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 <sub>IL</sub> )
1	0	1	Subsystem clock (f <sub>SUB</sub> )
Other than above			Setting prohibited

### 25.3.8 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)**.

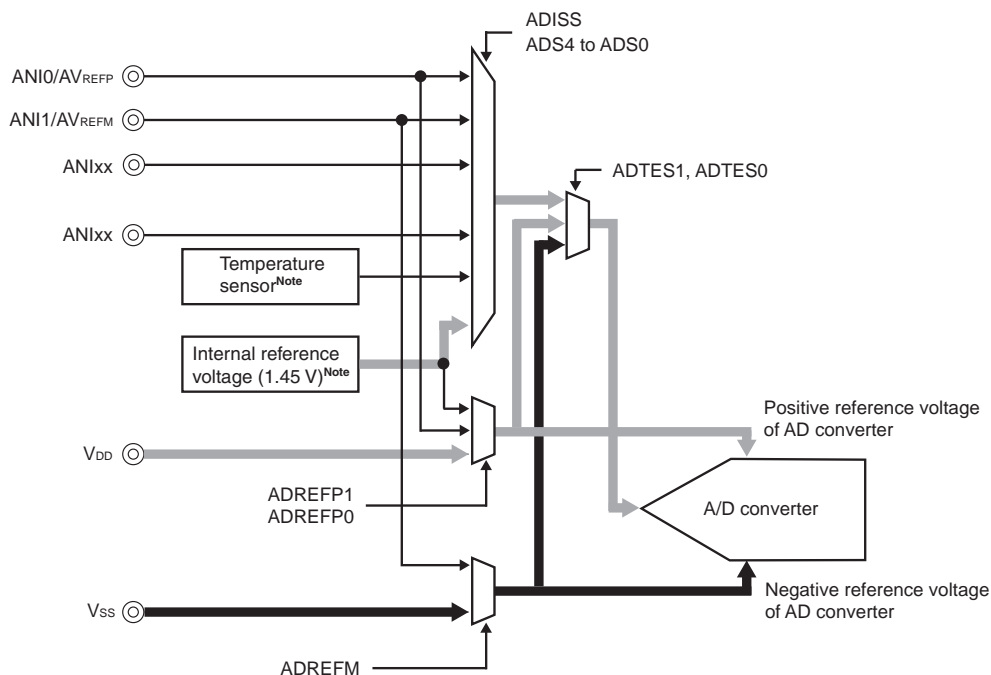
The analog multiplexer can be checked using the following procedure.

- <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 25-15. Configuration of A/D Test Function**



**Note** This setting can be used only in HS (high-speed main) mode.

**25.3.8.1 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 (ANLxx), 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 25-16. 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	ANLxx/temperature sensor output <sup>Note</sup> /internal reference voltage (1.45 V) <sup>Note</sup> (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) <sup>Note</sup>
Other than 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.

**25.3.8.2 Analog input channel specification register (ADS)**

This register specifies the input channel of the analog voltage to be A/D converted.

Set the A/D test register (ADTES) to 00H when measuring the ANLxx/temperature sensor output /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 25-17. 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

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P21/ANI0/AV <sub>REFFP</sub> pin
0	0	0	0	0	1	ANI1	P20/ANI1/AV <sub>REFM</sub> pin
0	1	0	0	0	0	ANI16	P22/ANI16 pin
0	1	0	0	0	1	ANI17	P23/ANI17 pin
0	1	0	0	1	0	ANI18	P24/ANI18 pin
0	1	0	0	1	1	ANI19	P25/ANI19 pin
0	1	0	1	0	0	ANI20	P26/ANI20 pin
0	1	0	1	0	1	ANI21	P27/ANI21 pin
0	1	0	1	1	0	ANI22	P10/ANI22 pin
0	1	0	1	1	1	ANI23	P11/ANI23 pin
0	1	1	0	0	0	ANI24	P12/ANI24 pin
0	1	1	0	0	1	ANI25	P13/ANI25 pin
1	0	0	0	0	0	–	Temperature sensor output voltage <sup>Note</sup>
1	0	0	0	0	1	–	Internal reference voltage (1.45 V) <sup>Note</sup>
Other than above						Setting prohibited	

**Note** 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 1 and 2 (PM1 and PM2).
  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 1, 2 (PMC1, PMC2).
  5. Only rewrite the value of the ADISS bit while conversion operation is stopped (ADCS = 0, ADCE = 0).
  6. If using AV<sub>REFFP</sub> 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<sub>REFM</sub> as the negative reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.
  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 12.7.4 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<sub>ADREF</sub>) shown in 32.3.2 or 33.3.2 Supply current characteristics is added.

**25.3.9 Digital output signal level detection function for I/O pins**

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 pins, the digital output level of the pin can be read when the pin is set to output mode.

**25.3.9.1 Port mode select register (PMS)**

This register is used to select the output level from output latch level or pin output level when the pin is output mode in which the PMm bit of the 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 this register to 00H.

**Figure 25-18. Format of Port Mode Select Register (PMS)**

Address: F007BH 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 (PMm = 0)
0	Pmn register value is read.
1	Digital output level of the pin is read.

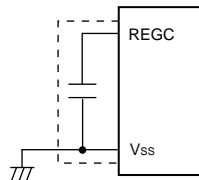
**Remark** m = 0 to 7, 12  
n = 0 to 7

- Cautions**
1. While the PMS0 bit of the PMS register is 1, do not change the value of the Px register by using a read-modify instruction. To change the value of the Px register, use an 8-bit memory manipulation instruction.
  2. PMS control cannot be used for the dedicated LCD pins and the input-only pins (P121 to P124 and P137).
  3. PMS control cannot be used for alternate-function pins being used as segment output pins. (“L” is always read when this register is read.)
  4. PMS control cannot be used for the P74 (TKBO00), P75 (TKBO01-2), P76 (TKBO01-1), and P77 (TKBO01-0) pins when they are being used as TMKB2 output pins and the forced output stop function is enabled (Hi-Z output only). (“L” is always read when this register is read.)
  5. PMS control cannot be used for P61 and P60 when IICA0EN (bit 4 of the PER0 register) is 0.

CHAPTER 26 REGULATOR

26.1 Regulator Overview

The RL78/L13 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<sub>SS</sub> 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 26-1.

Table 26-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
LV (low voltage main) mode	1.8 V	—
LS (low-speed main) mode		
HS (high-speed main) mode	1.8 V	In STOP mode
		When both the high-speed system clock (f <sub>MX</sub> ) and the high-speed on-chip oscillator clock (f <sub>IH</sub> ) are stopped during CPU operation with the subsystem clock (f <sub>XT</sub> )
	2.1 V	When both the high-speed system clock (f <sub>MX</sub> ) and the high-speed on-chip oscillator clock (f <sub>IH</sub> ) are stopped during the HALT mode when the CPU operation with the subsystem clock (f <sub>XT</sub> ) has been set
		Other than above (include during OCD mode) <sup>Note</sup>

**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 27 OPTION BYTE

### 27.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/L13 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.

#### 27.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)

##### (1) 000C0H/010C0H

- Operation of watchdog timer
  - Enabling or disabling of counter operation
  - Operation is stopped or enabled 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 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 32.4 or 33.4 AC Characteristics. This is done by utilizing the voltage detection circuit 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 detection circuit 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

<R> Make the setting depending on the main system clock frequency ( $f_{\text{MAIN}}$ ) and power supply voltage ( $V_{\text{DD}}$ ) to be used.

- LV (low voltage main) mode
- LS (low speed main) mode
- HS (high speed main) mode
- Setting of the frequency of the high-speed on-chip oscillator
  - Select from 48 MHz/24 MHz/16 MHz/12 MHz/8 MHz/6 MHz/4 MHz /3 MHz/2 MHz/1 MHz (TYP.).

**Caution** Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

**27.1.2 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.

## 27.2 Format of User Option Byte

The format of user option byte is shown below.

**Figure 27-1. Format of User Option Byte (000C0H/010C0H)**

Address: 000C0H/010C0H<sup>Note 1</sup>

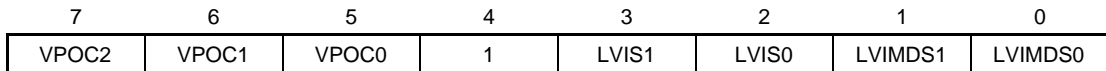
7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON
WDTINT	Use of interval interrupt of watchdog timer						
0	Interval interrupt is not used.						
1	Interval interrupt is generated when 75% + 1/2 <sub>f<sub>IL</sub></sub> of the overflow time is reached.						
WINDOW1	WINDOW0	Watchdog timer window open period <sup>Note 2</sup>					
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 <sub>IL</sub> = 17.25 kHz (MAX.))				
0	0	0	2 <sup>6</sup> /f <sub>IL</sub> (3.71 ms)				
0	0	1	2 <sup>7</sup> /f <sub>IL</sub> (7.42 ms)				
0	1	0	2 <sup>8</sup> /f <sub>IL</sub> (14.84 ms)				
0	1	1	2 <sup>9</sup> /f <sub>IL</sub> (29.68 ms)				
1	0	0	2 <sup>11</sup> /f <sub>IL</sub> (118.72 ms)				
1	0	1	2 <sup>13</sup> /f <sub>IL</sub> (474.90 ms)				
1	1	0	2 <sup>14</sup> /f <sub>IL</sub> (949.80 ms)				
1	1	1	2 <sup>16</sup> /f <sub>IL</sub> (3799.19 ms)				
WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)						
0	Counter operation stopped in HALT/STOP mode <sup>Note 2</sup>						
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<sub>IL</sub>: Low-speed on-chip oscillator clock frequency

Figure 27-2. Format of User Option Byte (000C1H/010C1H) (1/2)

Address: 000C1H/010C1H<sup>Note</sup>



- LVD setting (interrupt & reset mode)

Detection voltage			Option byte setting value												
V <sub>LV<sub>DH</sub></sub>		V <sub>LV<sub>DL</sub></sub>	Mode setting		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0						
Rising edge	Falling edge	Falling edge	LVIMDS1	LVIMDS0											
1.77 V	1.73 V	1.63 V	1	0	0	0	0	1	0						
1.88 V	1.84 V							0	1						
2.92 V	2.86 V							0	0						
1.98 V	1.94 V	1.84 V			1	0	0	0	1	1	0				
2.09 V	2.04 V									0	1				
3.13 V	3.06 V									0	0				
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							0	1	0	1	1	1	0
3.02 V	2.96 V													0	1
4.06 V	3.98 V													0	0
Other than above			Setting prohibited												

- LVD setting (reset mode)

Detection voltage		Option byte setting value								
V <sub>LV<sub>D</sub></sub>		Mode setting		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0		
Rising edge	Falling edge	LVIMDS1	LVIMDS0							
1.67 V	1.63 V	1	1	0	0	0	1	1		
1.77 V	1.73 V			0	0	0	1	0		
1.88 V	1.84 V			0	0	1	1	1		
1.98 V	1.94 V			0	0	1	1	0		
2.09 V	2.04 V			0	0	1	0	1		
2.50 V	2.45 V			0	1	0	1	1		
2.61 V	2.55 V			0	1	0	1	0		
2.71 V	2.65 V			0	1	0	0	1		
2.81 V	2.75 V			0	1	1	1	1		
2.92 V	2.86 V			0	1	1	1	0		
3.02 V	2.96 V			0	1	1	0	1		
3.13 V	3.06 V			0	0	1	0	0		
3.75 V	3.67 V			0	1	0	0	0		
4.06 V	3.98 V			0	1	1	0	0		
Other than above				Setting 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 24 VOLTAGE DETECTOR**.
  2. The detection voltage is a typical value. For details, see **32.6.5** or **33.6.5 LVD circuit characteristics**.

(Cautions are listed on the next page.)

Figure 27-2. Format of User Option Byte (000C1H/010C1H) (2/2)

Address: 000C1H/010C1H<sup>Note</sup>

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 <sub>LVD</sub>		Mode setting		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
Rising edge	Falling edge	LVIMDS1	LVIMDS0					
1.67 V	1.63 V	0	1	0	0	0	1	1
1.77 V	1.73 V			0	0	0	1	0
1.88 V	1.84 V			0	0	1	1	1
1.98 V	1.94 V			0	0	1	1	0
2.09 V	2.04 V			0	0	1	0	1
2.50 V	2.45 V			0	1	0	1	1
2.61 V	2.55 V			0	1	0	1	0
2.71 V	2.65 V			0	1	0	0	1
2.81 V	2.75 V			0	1	1	1	1
2.92 V	2.86 V			0	1	1	1	0
3.02 V	2.96 V			0	1	1	0	1
3.13 V	3.06 V			0	0	1	0	0
3.75 V	3.67 V			0	1	0	0	0
4.06 V	3.98 V			0	1	1	0	0
Other than above	Setting prohibited							

• LVD setting (LVDOFF)

Detection voltage		Option byte setting value						
V <sub>LVD</sub>		Mode setting		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
Rising edge	Falling edge	LVIMDS1	LVIMDS0					
-	-	×	1	1	×	×	×	×
Other than above	Setting 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 32.4 or 33.4 AC Characteristics. This is done by utilizing the voltage detection circuit 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 detection circuit 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. x: don't care

2. For details on the LVD circuit, see CHAPTER 24 VOLTAGE DETECTOR.
3. The detection voltage is a typical value. For details, see 32.6.5 or 33.6.5 LVD circuit characteristics.



Figure 27-3. Format of Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H<sup>Note</sup>

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 <sub>MAIN</sub> )	Operating voltage range (V <sub>DD</sub> )
0	0	LV (low voltage main) mode	1 to 4 MHz	1.6 to 5.5 V
1	0	LS (low speed main) mode	1 to 8 MHz	1.8 to 5.5 V
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 clock	
					f <sub>HOCO</sub>	f <sub>IH</sub>
1	0	0	0	0	48 MHz	24 MHz
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	1 MHz
Other than above					Setting prohibited	

**Note** Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

**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 32.4 or 33.4 AC Characteristics.

### 27.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

**Figure 27-4. Format of On-chip Debug Option Byte (000C3H/010C3H)**

Address: 000C3H/010C3H<sup>Note</sup>

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 bits 6 to 1 to 000010B.**

**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 bits 3 to 1 to their default value (0, 1, and 0).

## 27.4 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 $2^9/f_{IL}$ , ; Stops watchdog timer operation during HALT/STOP mode
	DB	1AH	; Select 1.63 V for $V_{LVDL}$ ; Select rising edge 1.77 V, falling edge 1.73 V for $V_{LVDH}$ ; Select the interrupt & reset mode as the LVD operation mode
	DB	2DH	; Select the LV (low voltage main) mode as the flash operation mode and 1 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

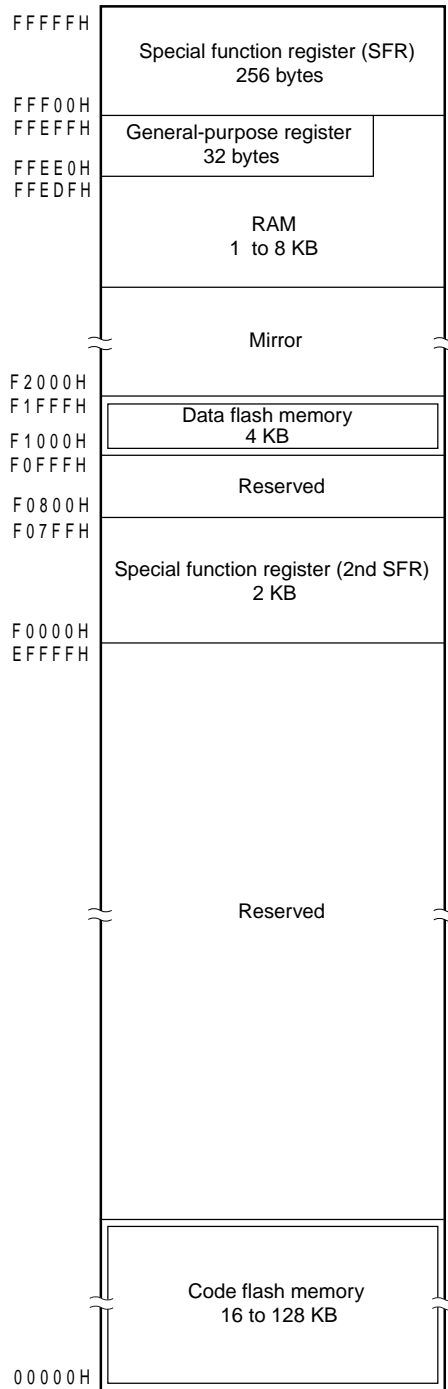
When the boot swap function is used during self programming, 000C0H to 000C3H is switched to 010C0H to 010C3H. Describe to 010C0H to 010C3H, therefore, the same values as 000C0H to 000C3H as follows.

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 $2^9/f_{IL}$ , ; Stops watchdog timer operation during HALT/STOP mode
	DB		1AH	; Select 1.63 V for $V_{LVDL}$ ; Select rising edge 1.77 V, falling edge 1.73 V for $V_{LVDH}$ ; Select the interrupt & reset mode as the LVD operation mode
	DB		2DH	; Select the LV (low main voltage) mode as the flash operation mode and 1 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 28 FLASH MEMORY**

The RL78 microcontroller incorporates the flash memory to which a program can be written, erased, and overwritten. 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 **28.1**)  
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 **28.2**)  
Data can be written to the flash memory on-board through UART communication with an external device (microcontroller or ASIC).
- Self-programming (see **28.5**)  
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 **28.7 Data Flash**.

## 28.1 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 Densai Machida Mfg. Co., Ltd.

**Table 28-1. Wiring Between RL78 Microcontroller and Dedicated Flash Memory Programmer**

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.	
					64-pin	
Signal Name		I/O	Pin Function		LQFP (12×12), LFQFP (10×10)	LQFP (14×14), LFQFP (12×12)
PG-FP5, FL-PR5	E1 On-chip Debugging Emulator					
–	TOOL0	I/O	Transmit/receive signal	TOOL0/P40	5	9
SI/RxD	–	I/O	Transmit/receive signal			
–	RESET	Output	Reset signal	RESET	6	10
/RESET	–	Output				
V <sub>DD</sub>		I/O	V <sub>DD</sub> voltage generation/ power monitoring	V <sub>DD</sub>	14	18
GND		–	Ground	V <sub>SS</sub>	13	17
				EV <sub>SS</sub>	–	–
				REGC <sup>Note</sup>	12	16
FLMD1	EMV <sub>DD</sub>	–	Driving power for TOOL pin	V <sub>DD</sub>	14	18

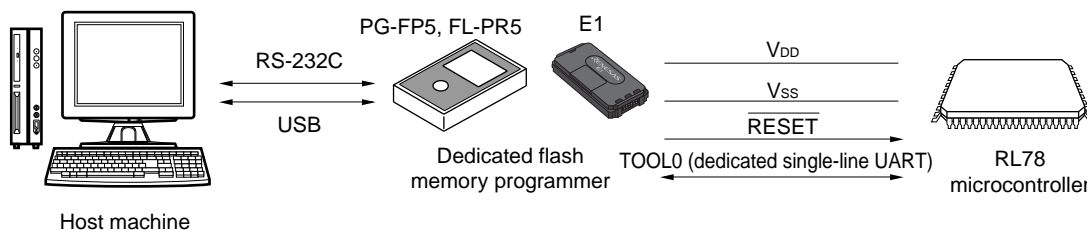
**Note** Connect REGC pin to ground via a capacitor (0.47 to 1  $\mu$ F).

**Remark** Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

**28.1.1 Programming environment**

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

**Figure 28-1. Environment for Writing Program to Flash Memory**



A host machine that controls the dedicated flash memory programmer is necessary.

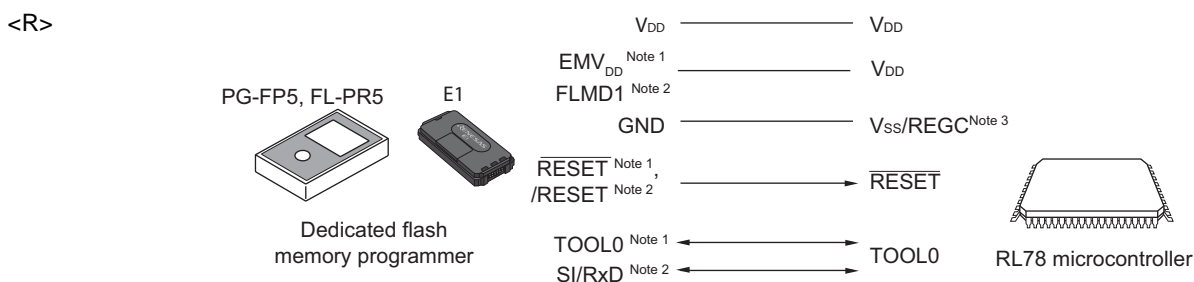
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

**28.1.2 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 28-2. 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  $\mu$ F).



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 28-2. Pin Connection**

Dedicated Flash Memory Programmer			RL78 Microcontroller
Signal Name		I/O	Pin Name <sup>Note 2</sup>
PG-FP5, FL-PR5	E1 On-chip Debugging Emulator		
V <sub>DD</sub>		I/O	V <sub>DD</sub> voltage generation/power monitoring
GND		–	Ground
FLMD1	EMV <sub>DD</sub>	–	Driving power for TOOL pin
/RESET	–	Output	Reset signal
–	RESET	Output	
–	TOOL0	I/O	Transmit/receive signal
SI/RxD	–	I/O	Transmit/receive signal

- Notes**
1. Connect REGC pin to ground via a capacitor (0.47 to 1  $\mu$ F).
  2. Pins to be connected differ with the product. For details, see **Table 28-1**.

### 28.2 Serial Programming Using External Device (that Incorporates UART)

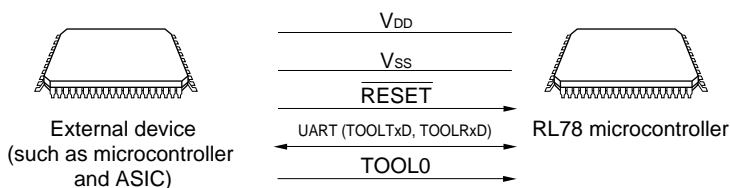
On-board data writing to the internal flash memory is possible by using the RL78 microcontroller and an external device (a microcontroller or ASIC) connected to a UART.

On the development of flash memory programmer by user, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

#### 28.2.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

**Figure 28-3. Environment for Writing Program to Flash Memory**



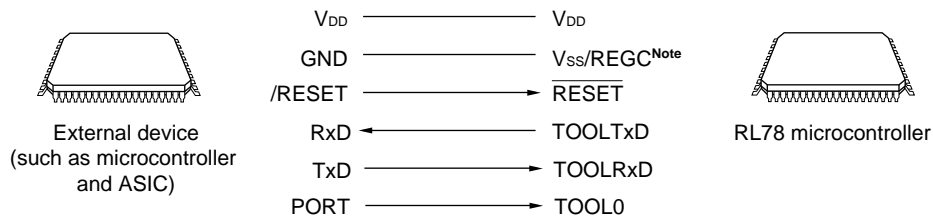
Processing to write data to or delete data from the RL78 microcontroller by using an external device is performed on-board. Off-board writing is not possible.

**28.2.2 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 28-4. Communication with External Device**



**Note** Connect REGC pin to ground via a capacitor (0.47 to 1  $\mu$ F).

The external device generates the following signals for the RL78 microcontroller.

**Table 28-3. Pin Connection**

External Device			RL78 Microcontroller
Signal Name	I/O	Pin Function	Pin Name <sup>Note 2</sup>
V <sub>DD</sub>	I/O	V <sub>DD</sub> voltage generation/power monitoring	V <sub>DD</sub>
GND	–	Ground	V <sub>SS</sub> , REGC <sup>Note 1</sup>
RESETOUT	Output	Reset signal output	$\overline{\text{RESET}}$
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  $\mu$ F).

### 28.3 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** Refer to flash programming mode, see **28.4.2 Flash memory programming mode**.

#### 28.3.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1 k $\Omega$  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. Furthermore, when this pin is used via pull-down resistors, use the 500 k $\Omega$  or more resistors.

When used as an output pin: When this pin is used via pull-down resistors, use the 500 k $\Omega$  or more resistors.

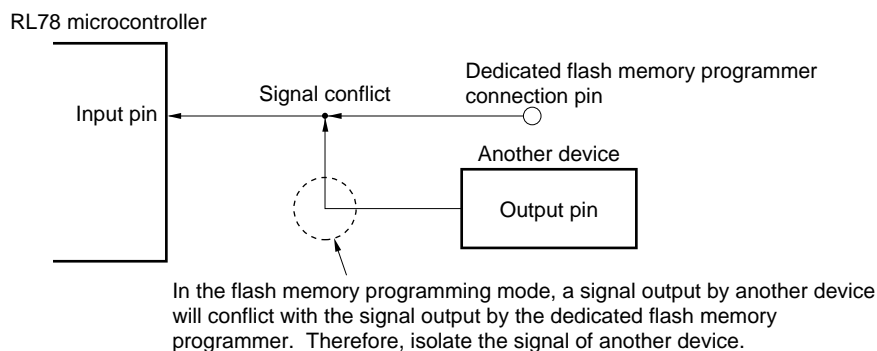
- Remarks**
1.  $t_{HD}$ : How long to keep the TOOL0 pin at the low level from when the external and internal resets end for setting of the flash memory programming mode (see **32.11** or **33.11 Timing Specs for Switching Flash Memory Programming Modes**)
  2. The SAU and IICA pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

#### 28.3.2 $\overline{\text{RESET}}$ 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 28-5. Signal Conflict ( $\overline{\text{RESET}}$  Pin)**



### 28.3.3 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.

### 28.3.4 REGC pin

Connect the REGC pin to GND via a capacitor having excellent characteristics (0.47 to 1  $\mu\text{F}$ ) in the same manner as during normal operation. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

### 28.3.5 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 ( $f_{IH}$ ) is used.

### 28.3.6 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.

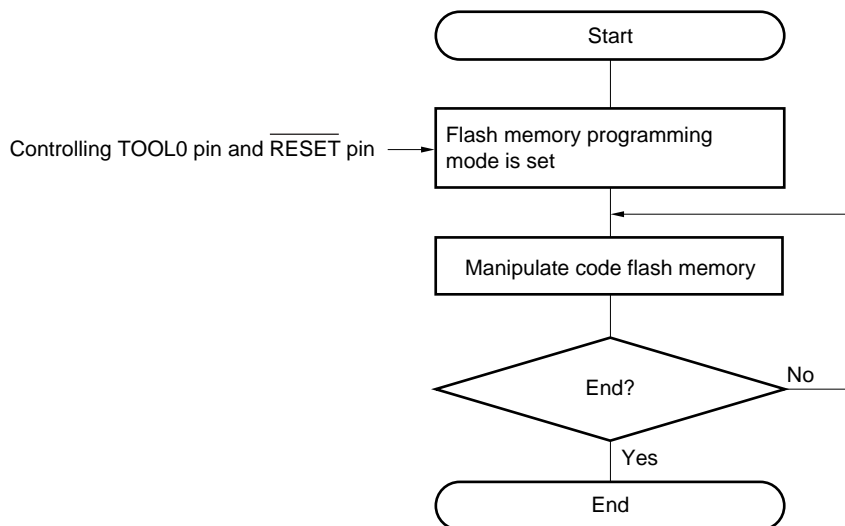
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the  $V_{DD}$  and  $V_{SS}$  pins to  $V_{DD}$  and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

## 28.4 Serial Programming Method

### 28.4.1 Controlling serial programming

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

**Figure 28-6. Code Flash Memory Manipulation Procedure**



### 28.4.2 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.

<When programming by using the dedicated flash memory programmer>

Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

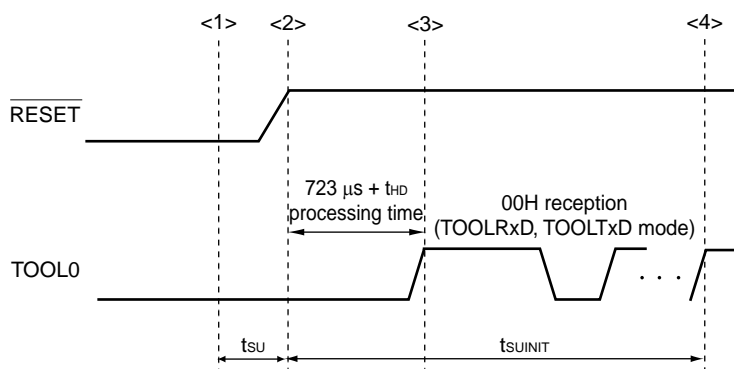
<Serial programming using an external device (UART communication)>

Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 28-4**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in **Figure 28-7**. For details, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

**Table 28-4. Relationship Between TOOL0 Pin and Operation Mode After Reset Release**

TOOL0	Operation Mode
V <sub>DD</sub>	Normal operation mode
0 V	Flash memory programming mode

**Figure 28-7. 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** *tsuINIT*: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the resets end.

*tsu*: How long from when the TOOL0 pin is placed at the low level until an external reset ends

*tHD*: 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 **32.11** or **33.11 Timing Specifications 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 28-5. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified**

Power Supply Voltage (V <sub>DD</sub> )	User Option Byte Setting for Switching to Flash Memory Programming Mode		Flash Programming Mode
	Flash Operation Mode	Operating Frequency	
2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	Blank state		Full speed mode
	HS (high speed main) mode	1 MHz to 24 MHz	Full speed mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low voltage main) mode	1 MHz to 4 MHz	Wide voltage mode
2.4 V ≤ V <sub>DD</sub> < 2.7 V	Blank state		Full speed mode
	HS (high speed main) mode	1 MHz to 16 MHz	Full speed mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low voltage main) mode	1 MHz to 4 MHz	Wide voltage mode
1.8 V ≤ V <sub>DD</sub> < 2.4 V	Blank state		Wide voltage mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low voltage main) mode	1 MHz to 4 MHz	Wide voltage mode

- Remarks**
- Using both the wide voltage mode and full speed mode imposes no restrictions on writing, deletion, or verification.
  - For details about communication commands, see **28.4.4 Communication commands**.

**28.4.3 Selecting communication mode**

Communication mode of the RL78 microcontroller as follows.

**Table 28-6. Communication Modes**

Communication Mode	Standard Setting <sup>Note 1</sup>				Pins Used
	Port	Speed <sup>Note 2</sup>	Frequency	Multiply Rate	
1-line mode (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**
- Selection items for Standard settings on GUI of the flash memory programmer.
  - 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.

#### 28.4.4 Communication commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 28-7**.

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 28-7. 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 <sup>Note</sup> .
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 28-8** is a list of signature data and **Table 28-9** shows an example of signature data.

**Table 28-8. 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, FFH, 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 28-9. 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	RSF10WLE	10 bytes	52 = "R" 35 = "5" 46 = "F" 31 = "1" 30 = "0" 57 = "W" 4C = "L" 45 = "E" 20 = " " 20 = " "
Code flash memory area last address	Code flash memory area 00000H to 0FFFFH (64 KB)	3 bytes	FF FF 00
Data flash memory area last address	Data flash memory area F1000H to F1FFFH (4 KB)	3 bytes	FF 1F 0F
Firmware version	Ver.1.23	3 bytes	01 02 03

## 28.5 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 (HIOSSTOP = 0). The flash self-programming library should be executed after 30  $\mu$ s have elapsed when FRQSEL4 of the user option byte (000C2H) is 0 or 80  $\mu$ s have elapsed when FRQSEL4 is 1.

- Remarks**
1. For details of the self-programming function, refer to the **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.

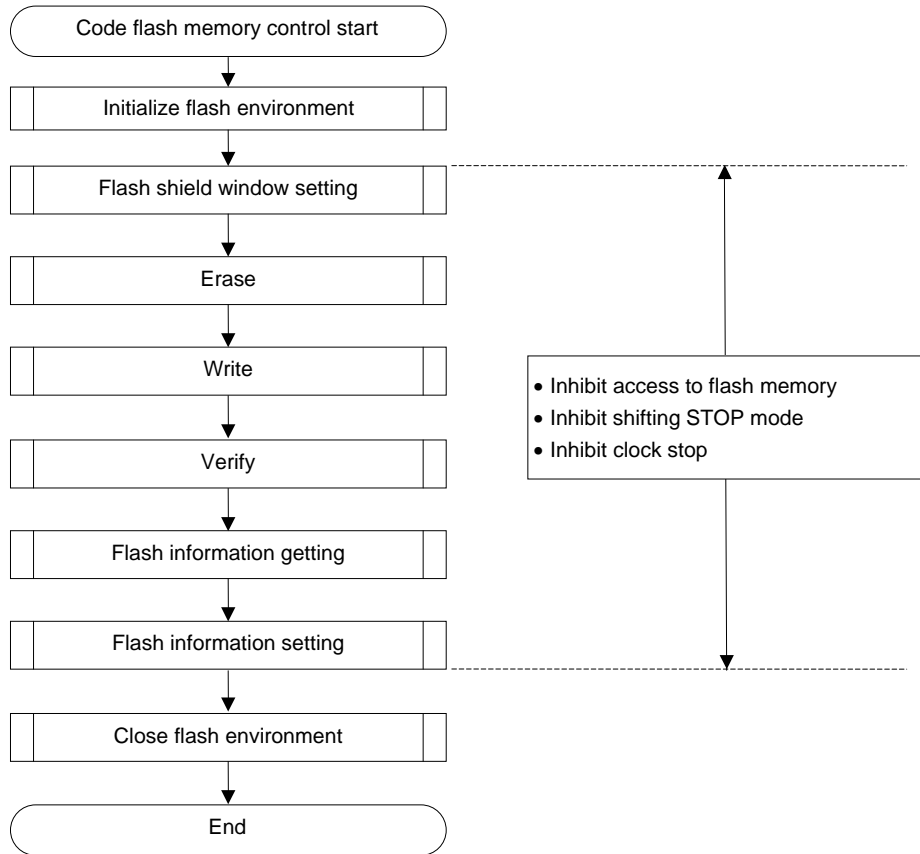
If the argument `fsl_flash_voltage_u08` is 00H when the `FSL_Init` function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

- Remark** Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.

**28.5.1 Self-programming procedure**

The following figure illustrates a flow for rewriting the code flash memory by using a flash self-programming library.

**Figure 28-8. Flow of Self Programming (Rewriting Flash Memory)**



**28.5.2 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.

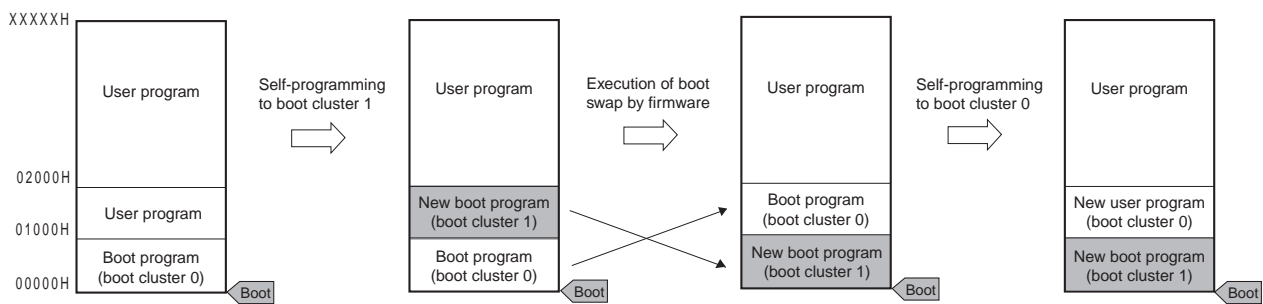
The boot swap function is used to avoid this problem.

Before erasing boot cluster 0<sup>Note</sup>, 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 28-9. Boot Swap Function**

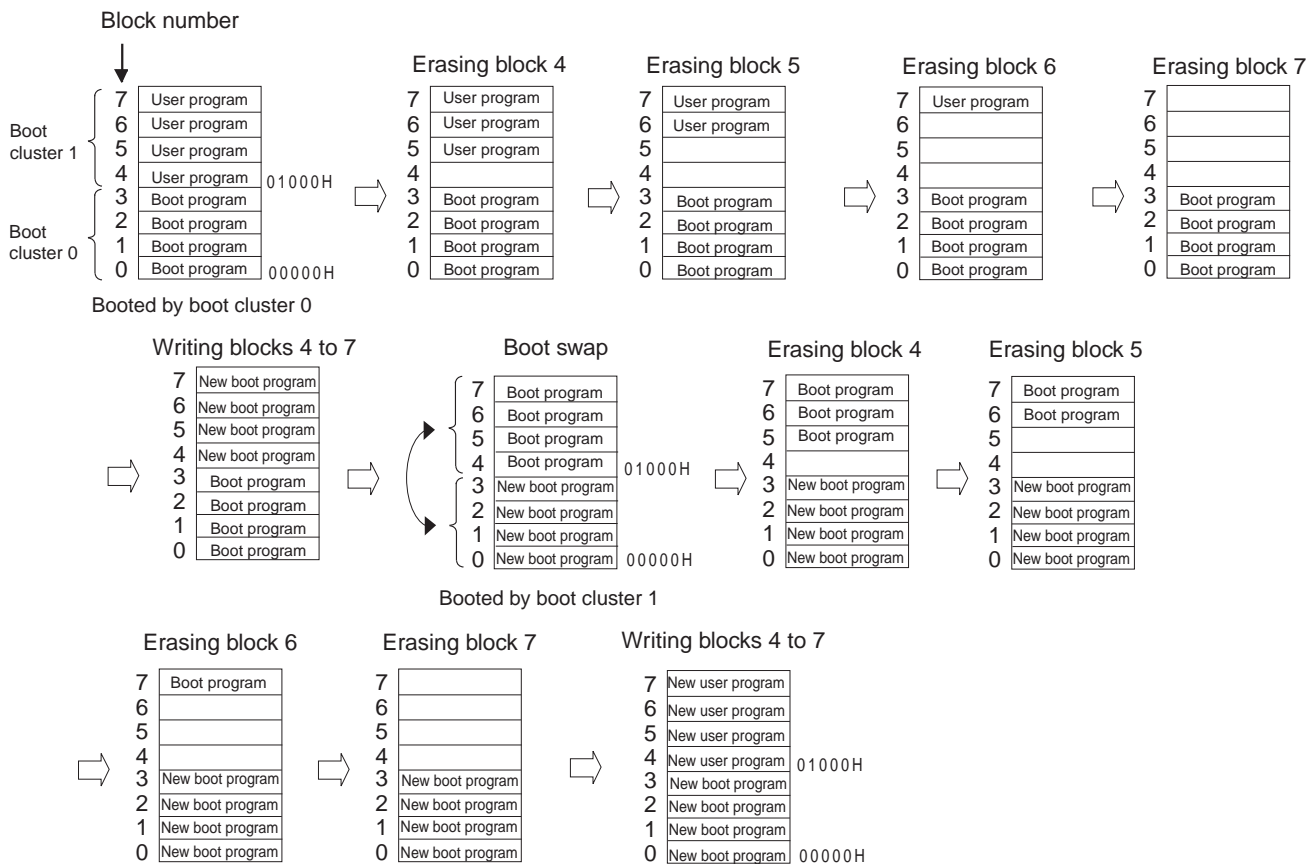


In an example of above figure, it is as follows.

Boot cluster 0: Boot area before boot swap

Boot cluster 1: Boot area after boot swap

Figure 28-10. Example of Executing Boot Swapping



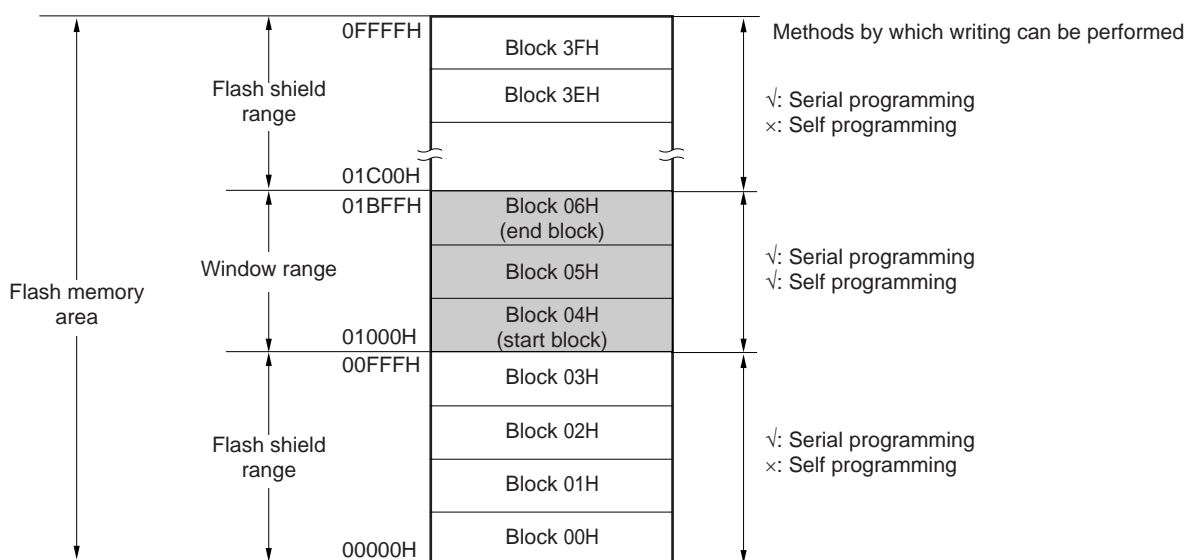
**28.5.3 Flash shield window function**

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 28-11. Flash Shield Window Setting Example**  
**(Target Devices: R5F10WLE, 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 28-10. 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 28.6 Security Settings to prohibit writing/erasing during serial programming.

## 28.6 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 flash memory is prohibited during serial programming. However, blocks can be written by means of self programming.

After the security settings are specified, releasing the security settings 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 28-11 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 **28.5.3** for detail).

**Table 28-11. 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. <sup>Note</sup>
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 28.5.3 for detail).

**Table 28-12. 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.



## 28.7 Data Flash

### 28.7.1 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 the **RL78 Family Data Flash Library User's Manual**.
- The data flash memory can also be rewritten through serial programming using the dedicated flash memory programmer or an external device.
- The data flash can be erased in 1-block (1 KB) 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 HALT/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  $\mu$ s have elapsed when FRQSEL4 of the user option byte (000C2H) is 0 or 80  $\mu$ s have elapsed when FRQSEL4 is 1.**

**Remark** For details of flash programming mode, see **28.5 Self-Programming**.

## 28.7.2 Register controlling data flash memory

### 28.7.2.1 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 signal generation sets this register to 00H.

**Figure 28-12. 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.

### 28.7.3 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> Write 1 to bit 0 (DFLEN) of the data flash control register (DFLCTL).
- <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  $\mu$ s
  - LS (Low-speed main): 720 ns
  - LV (Low-voltage main): 10  $\mu$ s
- <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 flash data library should be executed after 30  $\mu$ s have elapsed when FRQSEL4 of the user option byte (000C2H) is 0 or 80  $\mu$ s have elapsed when FRQSEL4 is 1.

After initialization, the data flash memory can be read or rewritten by using a CPU instruction or flash data library. If the DMA controller operates when the data flash memory is accessed, follow either of these procedures:

#### (A) Hold pending/forced termination of DMA transfer

Before reading the data flash memory, hold DMA transfer pending of all the channels in use. Note that, wait for at least 3 clocks ( $f_{CLK}$ ) after setting the DWAITn bit to 1 and before reading the data flash memory. After reading the data flash, cancel holding of DMA transfer pending by clearing the DWAITn bit to 0, or before reading the data flash memory, forcibly terminate DMA transfer in accordance with **18.5.5 Forced termination by software**. Resume DMA transfer after reading the data flash memory.

#### (B) Accessing data flash memory by using library

Access the data flash memory by using the latest flash data library.

#### (C) Inserting NOP

Insert a NOP instruction immediately before the instruction that reads the data flash memory.

<Example>

MOVW HL, laddr16 ; Reads RAM.

NOP ; Insert NOP instruction before reading the data flash memory.

MOV A,[DE] ; Reads the data flash memory.

If a high-level language such as C is used, however, the compiler may generate two instructions per code. In this case, read the data flash memory in procedure (A) or (B) because the NOP instruction is not inserted immediately before the data flash reading instruction.

- Remarks**
1. DWAITn: Bit 4 of DMA mode control register n (DMCn)
  2.  $f_{CLK}$ : CPU/peripheral hardware clock frequency

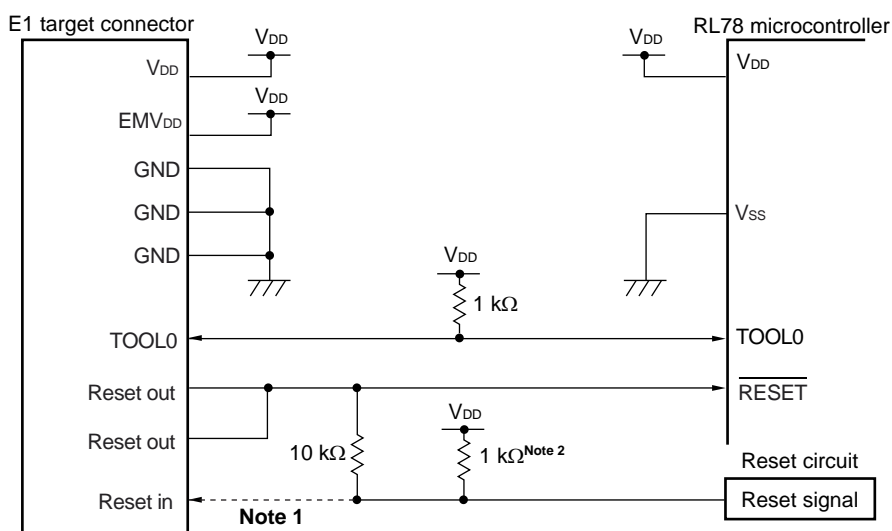
CHAPTER 29 ON-CHIP DEBUG FUNCTION

29.1 Connecting E1 On-chip Debugging Emulator

The RL78 microcontroller uses the V<sub>DD</sub>,  $\overline{\text{RESET}}$ , TOOL0, and V<sub>SS</sub> 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 29-1. Connection Example of E1 On-chip Debugging Emulator



- Notes**
1. Connecting the dotted line is not necessary during flash 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 O.D. buffer (output resistor: 100 Ω or less)

### 29.2 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 27 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 29-1. On-chip Debug Security ID**

Address	On-chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes <sup>Note</sup> (excluding all FFH)
010C4H to 010CDH	

<R>

<R>

Note. The setting FFFFFFFFFFFFFFFFFFFFFFH for the ID code is not possible.

### 29.3 Securing of User Resources

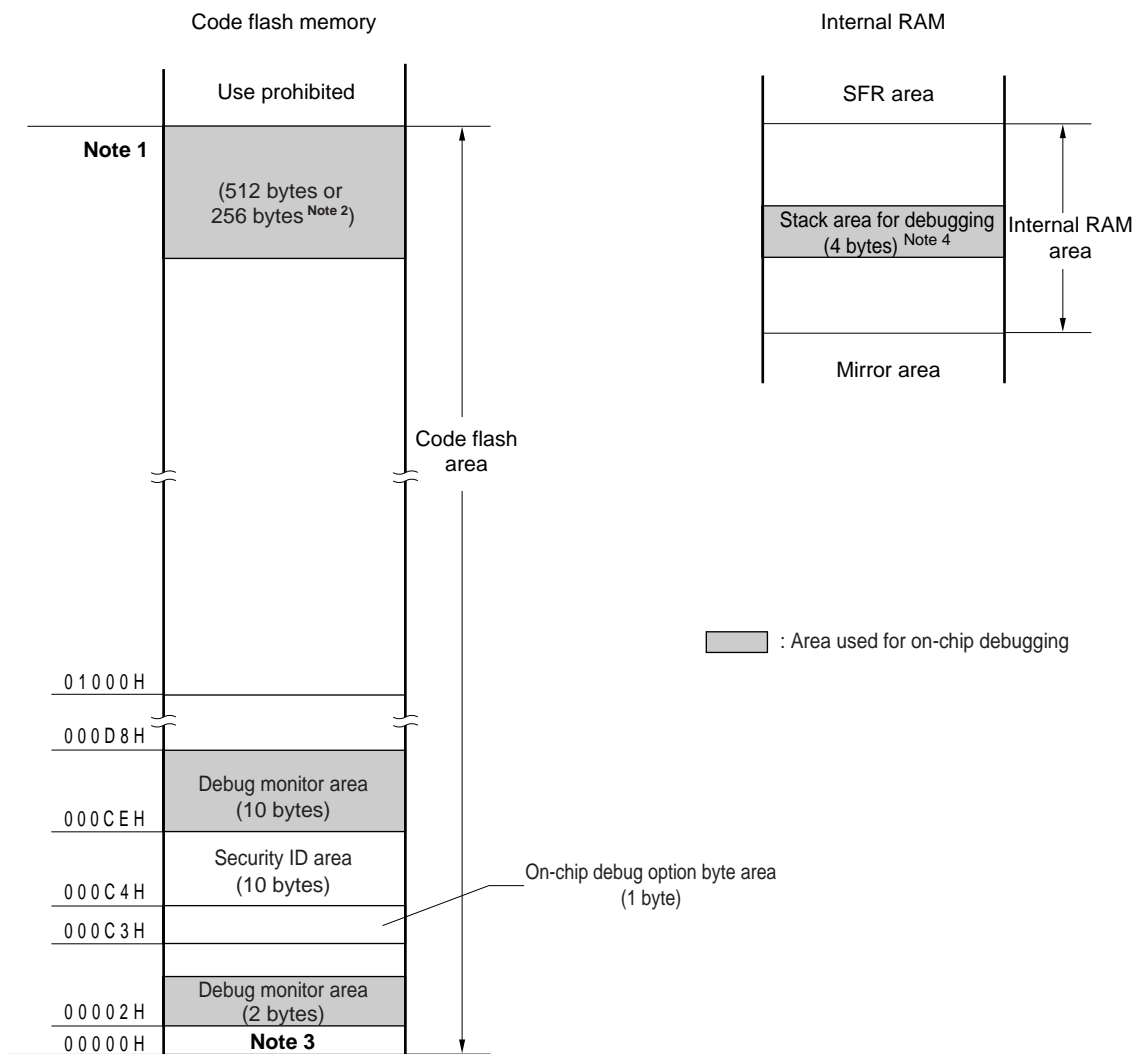
To perform communication between the RL78 microcontroller and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

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 29-2 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 29-2. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on products as follows.

Products (code flash memory capacity)	Address of Note 1
R5F10WLA, R5F10WMA	03FFFH
R5F10WLC, R5F10WMC	07FFFH
R5F10WLD, R5F10WMD	0BFFFH
R5F10WLE, R5F10WME	0FFFFH
R5F10WLF, R5F10WMF	17FFFH
R5F10WLG, R5F10WMG	1FFFFH

- When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.
- In debugging, reset vector is rewritten to address allocated to a monitor program.
- 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 30 BCD CORRECTION CIRCUIT

### 30.1 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).

### 30.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

#### 30.2.1 BCD correction result register (BCDADJ)

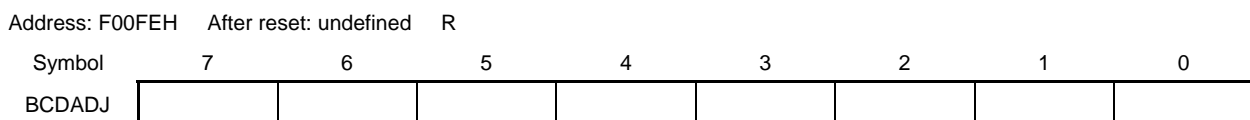
The BCDADJ register stores correction values for obtaining the add/subtract result as BCD code through add/subtract instructions using the A register as the operand.

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 30-1. Format of BCD Correction Result Register (BCDADJ)**



### 30.3 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 31 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)**.

## 31.1 Conventions Used in Operation List

### 31.1.1 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 31-1. Operand Identifiers and Specification Methods**

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol (SFR symbol) FFF00H to FFFFFH
sfrp	Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only <sup>Note</sup> ) FFF00H to FFFFFH
saddr	FFE20H to FFF1FH Immediate data or labels
saddrp	FFE20H to FF1FH Immediate data or labels (even addresses only <sup>Note</sup> )
addr20	00000H to FFFFFH Immediate data or labels
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions <sup>Note</sup> )
addr5	0080H to 00BFH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
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.

### 31.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

**Table 31-2. 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 <sub>H</sub> , X <sub>L</sub>	16-bit registers: X <sub>H</sub> = higher 8 bits, X <sub>L</sub> = lower 8 bits
X <sub>S</sub> , X <sub>H</sub> , X <sub>L</sub>	20-bit registers: X <sub>S</sub> = (bits 19 to 16), X <sub>H</sub> = (bits 15 to 8), X <sub>L</sub> = (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)

### 31.1.3 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 31-3. 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

### 31.1.4 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 31-4. 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.

31.2 Operation List

Table 31-5. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operands	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	x	x	x
		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 <small>Note 3</small>	1	1	–	A ← r			
		r, A <small>Note 3</small>	1	1	–	r ← A			
		A, PSW	2	1	–	A ← PSW			
		PSW, A	2	3	–	PSW ← A	x	x	x
		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<sub>CLK</sub>) 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<sub>CLK</sub>) when the code flash area is accessed, or when the data flash area 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 31-5. 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	–	$A \leftarrow \text{sfr}$			
		sfr, A	2	1	–	$\text{sfr} \leftarrow A$			
		A, [DE]	1	1	4	$A \leftarrow (\text{DE})$			
		[DE], A	1	1	–	$(\text{DE}) \leftarrow A$			
		A, ES:[DE]	2	2	5	$A \leftarrow (\text{ES}, \text{DE})$			
		ES:[DE], A	2	2	–	$(\text{ES}, \text{DE}) \leftarrow A$			
		A, [HL]	1	1	4	$A \leftarrow (\text{HL})$			
		[HL], A	1	1	–	$(\text{HL}) \leftarrow A$			
		A, ES:[HL]	2	2	5	$A \leftarrow (\text{ES}, \text{HL})$			
		ES:[HL], A	2	2	–	$(\text{ES}, \text{HL}) \leftarrow A$			
		A, [DE+byte]	2	1	4	$A \leftarrow (\text{DE} + \text{byte})$			
		[DE+byte], A	2	1	–	$(\text{DE} + \text{byte}) \leftarrow A$			
		A, ES:[DE+byte]	3	2	5	$A \leftarrow ((\text{ES}, \text{DE}) + \text{byte})$			
		ES:[DE+byte], A	3	2	–	$((\text{ES}, \text{DE}) + \text{byte}) \leftarrow A$			
		A, [HL+byte]	2	1	4	$A \leftarrow (\text{HL} + \text{byte})$			
		[HL+byte], A	2	1	–	$(\text{HL} + \text{byte}) \leftarrow A$			
		A, ES:[HL+byte]	3	2	5	$A \leftarrow ((\text{ES}, \text{HL}) + \text{byte})$			
		ES:[HL+byte], A	3	2	–	$((\text{ES}, \text{HL}) + \text{byte}) \leftarrow A$			
		A, [SP+byte]	2	1	–	$A \leftarrow (\text{SP} + \text{byte})$			
		[SP+byte], A	2	1	–	$(\text{SP} + \text{byte}) \leftarrow A$			
		A, word[B]	3	1	4	$A \leftarrow (\text{B} + \text{word})$			
		word[B], A	3	1	–	$(\text{B} + \text{word}) \leftarrow A$			
		A, ES:word[B]	4	2	5	$A \leftarrow ((\text{ES}, \text{B}) + \text{word})$			
		ES:word[B], A	4	2	–	$((\text{ES}, \text{B}) + \text{word}) \leftarrow A$			
		A, word[C]	3	1	4	$A \leftarrow (\text{C} + \text{word})$			
		word[C], A	3	1	–	$(\text{C} + \text{word}) \leftarrow A$			
		A, ES:word[C]	4	2	5	$A \leftarrow ((\text{ES}, \text{C}) + \text{word})$			
		ES:word[C], A	4	2	–	$((\text{ES}, \text{C}) + \text{word}) \leftarrow A$			
		A, word[BC]	3	1	4	$A \leftarrow (\text{BC} + \text{word})$			
		word[BC], A	3	1	–	$(\text{BC} + \text{word}) \leftarrow A$			
		A, ES:word[BC]	4	2	5	$A \leftarrow ((\text{ES}, \text{BC}) + \text{word})$			
		ES:word[BC], A	4	2	–	$((\text{ES}, \text{BC}) + \text{word}) \leftarrow A$			

- Notes 1.** Number of CPU clocks ( $f_{\text{CLK}}$ ) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.
- 2.** Number of CPU clocks ( $f_{\text{CLK}}$ ) when the code flash area is accessed, or when the data flash area 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 31-5. 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	$A \leftarrow (HL + B)$			
		[HL+B], A	2	1	-	$(HL + B) \leftarrow A$			
		A, ES:[HL+B]	3	2	5	$A \leftarrow ((ES, HL) + B)$			
		ES:[HL+B], A	3	2	-	$((ES, HL) + B) \leftarrow A$			
		A, [HL+C]	2	1	4	$A \leftarrow (HL + C)$			
		[HL+C], A	2	1	-	$(HL + C) \leftarrow A$			
		A, ES:[HL+C]	3	2	5	$A \leftarrow ((ES, HL) + C)$			
		ES:[HL+C], A	3	2	-	$((ES, HL) + C) \leftarrow A$			
		X, laddr16	3	1	4	$X \leftarrow (addr16)$			
		X, ES:laddr16	4	2	5	$X \leftarrow (ES, addr16)$			
		X, saddr	2	1	-	$X \leftarrow (saddr)$			
		B, laddr16	3	1	4	$B \leftarrow (addr16)$			
		B, ES:laddr16	4	2	5	$B \leftarrow (ES, addr16)$			
		B, saddr	2	1	-	$B \leftarrow (saddr)$			
		C, laddr16	3	1	4	$C \leftarrow (addr16)$			
		C, ES:laddr16	4	2	5	$C \leftarrow (ES, addr16)$			
		C, saddr	2	1	-	$C \leftarrow (saddr)$			
		ES, saddr	3	1	-	$ES \leftarrow (saddr)$			
	XCH	A, r <sup>Note 3</sup>	1 (r = X) 2 (other than r = X)	1	-	$A \leftrightarrow r$			
		A, laddr16	4	2	-	$A \leftrightarrow (addr16)$			
		A, ES:laddr16	5	3	-	$A \leftrightarrow (ES, addr16)$			
		A, saddr	3	2	-	$A \leftrightarrow (saddr)$			
		A, sfr	3	2	-	$A \leftrightarrow sfr$			
		A, [DE]	2	2	-	$A \leftrightarrow (DE)$			
		A, ES:[DE]	3	3	-	$A \leftrightarrow (ES, DE)$			
		A, [HL]	2	2	-	$A \leftrightarrow (HL)$			
		A, ES:[HL]	3	3	-	$A \leftrightarrow (ES, HL)$			
		A, [DE+byte]	3	2	-	$A \leftrightarrow (DE + byte)$			
A, ES:[DE+byte]	4	3	-	$A \leftrightarrow ((ES, DE) + byte)$					
A, [HL+byte]	3	2	-	$A \leftrightarrow (HL + byte)$					
A, ES:[HL+byte]	4	3	-	$A \leftrightarrow ((ES, HL) + byte)$					

- 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 area is accessed, or when the data flash area 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 31-5. 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 $\leftrightarrow$ (HL+B)				
		A, ES:[HL+B]	3	3	–	A $\leftrightarrow$ ((ES, HL)+B)				
		A, [HL+C]	2	2	–	A $\leftrightarrow$ (HL+C)				
		A, ES:[HL+C]	3	3	–	A $\leftrightarrow$ ((ES, HL)+C)				
	ONEB	A	1	1	–	A $\leftarrow$ 01H				
		X	1	1	–	X $\leftarrow$ 01H				
		B	1	1	–	B $\leftarrow$ 01H				
		C	1	1	–	C $\leftarrow$ 01H				
		!addr16	3	1	–	(addr16) $\leftarrow$ 01H				
		ES:!addr16	4	2	–	(ES, addr16) $\leftarrow$ 01H				
		saddr	2	1	–	(saddr) $\leftarrow$ 01H				
	CLR B	A	1	1	–	A $\leftarrow$ 00H				
		X	1	1	–	X $\leftarrow$ 00H				
		B	1	1	–	B $\leftarrow$ 00H				
		C	1	1	–	C $\leftarrow$ 00H				
		!addr16	3	1	–	(addr16) $\leftarrow$ 00H				
		ES:!addr16	4	2	–	(ES, addr16) $\leftarrow$ 00H				
		saddr	2	1	–	(saddr) $\leftarrow$ 00H				
	MOVS	[HL+byte], X	3	1	–	(HL+byte) $\leftarrow$ X	x		x	
		ES:[HL+byte], X	4	2	–	(ES, HL+byte) $\leftarrow$ X	x		x	
	16-bit data transfer	MOVW	rp, #word	3	1	–	rp $\leftarrow$ word			
			saddrp, #word	4	1	–	(saddrp) $\leftarrow$ word			
sfrp, #word			4	1	–	sfrp $\leftarrow$ word				
AX, rp <sup>Note 3</sup>			1	1	–	AX $\leftarrow$ rp				
rp, AX <sup>Note 3</sup>			1	1	–	rp $\leftarrow$ AX				
AX, !addr16			3	1	4	AX $\leftarrow$ (addr16)				
!addr16, AX			3	1	–	(addr16) $\leftarrow$ AX				
AX, ES:!addr16			4	2	5	AX $\leftarrow$ (ES, addr16)				
ES:!addr16, AX			4	2	–	(ES, addr16) $\leftarrow$ AX				
AX, saddrp			2	1	–	AX $\leftarrow$ (saddrp)				
saddrp, AX			2	1	–	(saddrp) $\leftarrow$ AX				
AX, sfrp			2	1	–	AX $\leftarrow$ sfrp				
sfrp, AX			2	1	–	sfrp $\leftarrow$ 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 area is accessed, or when the data flash area 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 31-5. 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 area is accessed, or when the data flash area 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 31-5. 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 <sup>Note 3</sup>	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	x	x	x
		saddr, #byte	3	2	–	(saddr), CY ← (saddr)+byte	x	x	x
		A, r <sup>Note 4</sup>	2	1	–	A, CY ← A + r	x	x	x
		r, A	2	1	–	r, CY ← r + A	x	x	x
		A, !addr16	3	1	4	A, CY ← A + (addr16)	x	x	x
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	x	x	x
		A, saddr	2	1	–	A, CY ← A + (saddr)	x	x	x
		A, [HL]	1	1	4	A, CY ← A + (HL)	x	x	x
		A, ES:[HL]	2	2	5	A, CY ← A + (ES, HL)	x	x	x
		A, [HL+byte]	2	1	4	A, CY ← A + (HL+byte)	x	x	x
		A, ES:[HL+byte]	3	2	5	A, CY ← A + ((ES, HL)+byte)	x	x	x
		A, [HL+B]	2	1	4	A, CY ← A + (HL+B)	x	x	x
		A, ES:[HL+B]	3	2	5	A, CY ← A + ((ES, HL)+B)	x	x	x
		A, [HL+C]	2	1	4	A, CY ← A + (HL+C)	x	x	x
A, ES:[HL+C]	3	2	5	A, CY ← A + ((ES, HL) + C)	x	x	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 area is accessed, or when the data flash area 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 31-5. 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	x	x	x
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) +byte+CY	x	x	x
		A, r <sup>Note 3</sup>	2	1	–	A, CY ← A + r + CY	x	x	x
		r, A	2	1	–	r, CY ← r + A + CY	x	x	x
		A, laddr16	3	1	4	A, CY ← A + (addr16)+CY	x	x	x
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)+CY	x	x	x
		A, saddr	2	1	–	A, CY ← A + (saddr)+CY	x	x	x
		A, [HL]	1	1	4	A, CY ← A+ (HL) + CY	x	x	x
		A, ES:[HL]	2	2	5	A,CY ← A+ (ES, HL) + CY	x	x	x
		A, [HL+byte]	2	1	4	A, CY ← A+ (HL+byte) + CY	x	x	x
		A, ES:[HL+byte]	3	2	5	A,CY ← A+ ((ES, HL)+byte) + CY	x	x	x
		A, [HL+B]	2	1	4	A, CY ← A+ (HL+B) +CY	x	x	x
		A, ES:[HL+B]	3	2	5	A,CY ← A+((ES, HL)+B)+CY	x	x	x
		A, [HL+C]	2	1	4	A, CY ← A+ (HL+C)+CY	x	x	x
	A, ES:[HL+C]	3	2	5	A,CY ← A+ ((ES, HL)+C)+CY	x	x	x	
	SUB	A, #byte	2	1	–	A, CY ← A – byte	x	x	x
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte	x	x	x
		A, r <sup>Note 3</sup>	2	1	–	A, CY ← A – r	x	x	x
		r, A	2	1	–	r, CY ← r – A	x	x	x
		A, laddr16	3	1	4	A, CY ← A – (addr16)	x	x	x
		A, ES:!addr16	4	2	5	A, CY ← A – (ES, addr16)	x	x	x
		A, saddr	2	1	–	A, CY ← A – (saddr)	x	x	x
		A, [HL]	1	1	4	A, CY ← A – (HL)	x	x	x
		A, ES:[HL]	2	2	5	A,CY ← A – (ES, HL)	x	x	x
		A, [HL+byte]	2	1	4	A, CY ← A – (HL+byte)	x	x	x
		A, ES:[HL+byte]	3	2	5	A,CY ← A – ((ES, HL)+byte)	x	x	x
		A, [HL+B]	2	1	4	A, CY ← A – (HL+B)	x	x	x
A, ES:[HL+B]		3	2	5	A,CY ← A – ((ES, HL)+B)	x	x	x	
A, [HL+C]	2	1	4	A, CY ← A – (HL+C)	x	x	x		
A, ES:[HL+C]	3	2	5	A,CY ← A – ((ES, HL)+C)	x	x	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 area is accessed, or when the data flash area 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 31-5. 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	x	x	x
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte – CY	x	x	x
		A, r <sup>Note 3</sup>	2	1	–	A, CY ← A – r – CY	x	x	x
		r, A	2	1	–	r, CY ← r – A – CY	x	x	x
		A, laddr16	3	1	4	A, CY ← A – (addr16) – CY	x	x	x
		A, ES:!addr16	4	2	5	A, CY ← A – (ES, addr16) – CY	x	x	x
		A, saddr	2	1	–	A, CY ← A – (saddr) – CY	x	x	x
		A, [HL]	1	1	4	A, CY ← A – (HL) – CY	x	x	x
		A, ES:[HL]	2	2	5	A, CY ← A – (ES, HL) – CY	x	x	x
		A, [HL+byte]	2	1	4	A, CY ← A – (HL+byte) – CY	x	x	x
		A, ES:[HL+byte]	3	2	5	A, CY ← A – ((ES, HL)+byte) – CY	x	x	x
		A, [HL+B]	2	1	4	A, CY ← A – (HL+B) – CY	x	x	x
		A, ES:[HL+B]	3	2	5	A, CY ← A – ((ES, HL)+B) – CY	x	x	x
		A, [HL+C]	2	1	4	A, CY ← A – (HL+C) – CY	x	x	x
	A, ES:[HL+C]	3	2	5	A, CY ← A – ((ES:HL)+C) – CY	x	x	x	
	AND	A, #byte	2	1	–	A ← A ∧ byte	x		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∧ byte	x		
		A, r <sup>Note 3</sup>	2	1	–	A ← A ∧ r	x		
		r, A	2	1	–	R ← r ∧ A	x		
		A, laddr16	3	1	4	A ← A ∧ (addr16)	x		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	x		
		A, saddr	2	1	–	A ← A ∧ (saddr)	x		
		A, [HL]	1	1	4	A ← A ∧ (HL)	x		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	x		
		A, [HL+byte]	2	1	4	A ← A ∧ (HL+byte)	x		
		A, ES:[HL+byte]	3	2	5	A ← A ∧ ((ES:HL)+byte)	x		
		A, [HL+B]	2	1	4	A ← A ∧ (HL+B)	x		
		A, ES:[HL+B]	3	2	5	A ← A ∧ ((ES:HL)+B)	x		
A, [HL+C]		2	1	4	A ← A ∧ (HL+C)	x			
A, ES:[HL+C]	3	2	5	A ← A ∧ ((ES:HL)+C)	x				

- Notes 1.** Number of CPU clocks (f<sub>CLK</sub>) 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<sub>CLK</sub>) when the code flash area is accessed, or when the data flash area 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 31-5. 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	–	$A \leftarrow A \vee \text{byte}$		x	
		saddr, #byte	3	2	–	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$		x	
		A, r <sup>Note 3</sup>	2	1	–	$A \leftarrow A \vee r$		x	
		r, A	2	1	–	$r \leftarrow r \vee A$		x	
		A, laddr16	3	1	4	$A \leftarrow A \vee (\text{addr}16)$		x	
		A, ES:!addr16	4	2	5	$A \leftarrow A \vee (\text{ES}:\text{addr}16)$		x	
		A, saddr	2	1	–	$A \leftarrow A \vee (\text{saddr})$		x	
		A, [HL]	1	1	4	$A \leftarrow A \vee (\text{H})$		x	
		A, ES:[HL]	2	2	5	$A \leftarrow A \vee (\text{ES}:\text{HL})$		x	
		A, [HL+byte]	2	1	4	$A \leftarrow A \vee (\text{HL}+\text{byte})$		x	
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \vee ((\text{ES}:\text{HL})+\text{byte})$		x	
		A, [HL+B]	2	1	4	$A \leftarrow A \vee (\text{HL}+\text{B})$		x	
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \vee ((\text{ES}:\text{HL})+\text{B})$		x	
		A, [HL+C]	2	1	4	$A \leftarrow A \vee (\text{HL}+\text{C})$		x	
	A, ES:[HL+C]	3	2	5	$A \leftarrow A \vee ((\text{ES}:\text{HL})+\text{C})$		x		
	XOR	A, #byte	2	1	–	$A \leftarrow A \oplus \text{byte}$		x	
		saddr, #byte	3	2	–	$(\text{saddr}) \leftarrow (\text{saddr}) \oplus \text{byte}$		x	
		A, r <sup>Note 3</sup>	2	1	–	$A \leftarrow A \oplus r$		x	
		r, A	2	1	–	$r \leftarrow r \oplus A$		x	
		A, laddr16	3	1	4	$A \leftarrow A \oplus (\text{addr}16)$		x	
		A, ES:!addr16	4	2	5	$A \leftarrow A \oplus (\text{ES}:\text{addr}16)$		x	
		A, saddr	2	1	–	$A \leftarrow A \oplus (\text{saddr})$		x	
		A, [HL]	1	1	4	$A \leftarrow A \oplus (\text{HL})$		x	
		A, ES:[HL]	2	2	5	$A \leftarrow A \oplus (\text{ES}:\text{HL})$		x	
		A, [HL+byte]	2	1	4	$A \leftarrow A \oplus (\text{HL}+\text{byte})$		x	
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \oplus ((\text{ES}:\text{HL})+\text{byte})$		x	
		A, [HL+B]	2	1	4	$A \leftarrow A \oplus (\text{HL}+\text{B})$		x	
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \oplus ((\text{ES}:\text{HL})+\text{B})$		x	
A, [HL+C]		2	1	4	$A \leftarrow A \oplus (\text{HL}+\text{C})$		x		
A, ES:[HL+C]	3	2	5	$A \leftarrow A \oplus ((\text{ES}:\text{HL})+\text{C})$		x			

- Notes 1.** Number of CPU clocks ( $f_{\text{CLK}}$ ) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.
- 2.** Number of CPU clocks ( $f_{\text{CLK}}$ ) when the code flash area is accessed, or when the data flash area 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 31-5. 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	x	x	x
		laddr16, #byte	4	1	4	(addr16) – byte	x	x	x
		ES:laddr16, #byte	5	2	5	(ES:addr16) – byte	x	x	x
		saddr, #byte	3	1	–	(saddr) – byte	x	x	x
		A, r <sup>Note3</sup>	2	1	–	A – r	x	x	x
		r, A	2	1	–	r – A	x	x	x
		A, laddr16	3	1	4	A – (addr16)	x	x	x
		A, ES:laddr16	4	2	5	A – (ES:addr16)	x	x	x
		A, saddr	2	1	–	A – (saddr)	x	x	x
		A, [HL]	1	1	4	A – (HL)	x	x	x
		A, ES:[HL]	2	2	5	A – (ES:HL)	x	x	x
		A, [HL+byte]	2	1	4	A – (HL+byte)	x	x	x
		A, ES:[HL+byte]	3	2	5	A – ((ES:HL)+byte)	x	x	x
		A, [HL+B]	2	1	4	A – (HL+B)	x	x	x
		A, ES:[HL+B]	3	2	5	A – ((ES:HL)+B)	x	x	x
		A, [HL+C]	2	1	4	A – (HL+C)	x	x	x
	A, ES:[HL+C]	3	2	5	A – ((ES:HL)+C)	x	x	x	
	CMP0	A	1	1	–	A – 00H	x	0	0
		X	1	1	–	X – 00H	x	0	0
		B	1	1	–	B – 00H	x	0	0
		C	1	1	–	C – 00H	x	0	0
		laddr16	3	1	4	(addr16) – 00H	x	0	0
		ES:laddr16	4	2	5	(ES:addr16) – 00H	x	0	0
		saddr	2	1	–	(saddr) – 00H	x	0	0
	CMPS	X, [HL+byte]	3	1	4	X – (HL+byte)	x	x	x
		X, ES:[HL+byte]	4	2	5	X – ((ES:HL)+byte)	x	x	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 area is accessed, or when the data flash area 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 31-5. 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	x	x	x
		AX, AX	1	1	–	AX, CY ← AX+AX	x	x	x
		AX, BC	1	1	–	AX, CY ← AX+BC	x	x	x
		AX, DE	1	1	–	AX, CY ← AX+DE	x	x	x
		AX, HL	1	1	–	AX, CY ← AX+HL	x	x	x
		AX, !addr16	3	1	4	AX, CY ← AX+(addr16)	x	x	x
		AX, ES:!addr16	4	2	5	AX, CY ← AX+(ES:addr16)	x	x	x
		AX, saddrp	2	1	–	AX, CY ← AX+(saddrp)	x	x	x
		AX, [HL+byte]	3	1	4	AX, CY ← AX+(HL+byte)	x	x	x
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX+((ES:HL)+byte)	x	x	x
	SUBW	AX, #word	3	1	–	AX, CY ← AX – word	x	x	x
		AX, BC	1	1	–	AX, CY ← AX – BC	x	x	x
		AX, DE	1	1	–	AX, CY ← AX – DE	x	x	x
		AX, HL	1	1	–	AX, CY ← AX – HL	x	x	x
		AX, !addr16	3	1	4	AX, CY ← AX – (addr16)	x	x	x
		AX, ES:!addr16	4	2	5	AX, CY ← AX – (ES:addr16)	x	x	x
		AX, saddrp	2	1	–	AX, CY ← AX – (saddrp)	x	x	x
		AX, [HL+byte]	3	1	4	AX, CY ← AX – (HL+byte)	x	x	x
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX – ((ES:HL)+byte)	x	x	x
		CMPW	AX, #word	3	1	–	AX – word	x	x
	AX, BC		1	1	–	AX – BC	x	x	x
	AX, DE		1	1	–	AX – DE	x	x	x
	AX, HL		1	1	–	AX – HL	x	x	x
	AX, !addr16		3	1	4	AX – (addr16)	x	x	x
	AX, ES:!addr16		4	2	5	AX – (ES:addr16)	x	x	x
	AX, saddrp		2	1	–	AX – (saddrp)	x	x	x
	AX, [HL+byte]		3	1	4	AX – (HL+byte)	x	x	x
AX, ES: [HL+byte]	4		2	5	AX – ((ES:HL)+byte)	x	x	x	
Multiply	MULU	X	1	1	–	AX ← AxX			

**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 area is accessed, or when the data flash area 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 31-5. 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	–	$r \leftarrow r+1$	x	x	
		!addr16	3	2	–	$(addr16) \leftarrow (addr16)+1$	x	x	
		ES:!addr16	4	3	–	$(ES, addr16) \leftarrow (ES, addr16)+1$	x	x	
		saddr	2	2	–	$(saddr) \leftarrow (saddr)+1$	x	x	
		[HL+byte]	3	2	–	$(HL+byte) \leftarrow (HL+byte)+1$	x	x	
		ES: [HL+byte]	4	3	–	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)+1$	x	x	
	DEC	r	1	1	–	$r \leftarrow r-1$	x	x	
		!addr16	3	2	–	$(addr16) \leftarrow (addr16)-1$	x	x	
		ES:!addr16	4	3	–	$(ES, addr16) \leftarrow (ES, addr16)-1$	x	x	
		saddr	2	2	–	$(saddr) \leftarrow (saddr)-1$	x	x	
		[HL+byte]	3	2	–	$(HL+byte) \leftarrow (HL+byte)-1$	x	x	
		ES: [HL+byte]	4	3	–	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)-1$	x	x	
	INCW	rp	1	1	–	$rp \leftarrow rp+1$			
		!addr16	3	2	–	$(addr16) \leftarrow (addr16)+1$			
		ES:!addr16	4	3	–	$(ES, addr16) \leftarrow (ES, addr16)+1$			
		saddrp	2	2	–	$(saddrp) \leftarrow (saddrp)+1$			
		[HL+byte]	3	2	–	$(HL+byte) \leftarrow (HL+byte)+1$			
		ES: [HL+byte]	4	3	–	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)+1$			
	DECW	rp	1	1	–	$rp \leftarrow rp-1$			
		!addr16	3	2	–	$(addr16) \leftarrow (addr16)-1$			
		ES:!addr16	4	3	–	$(ES, addr16) \leftarrow (ES, addr16)-1$			
saddrp		2	2	–	$(saddrp) \leftarrow (saddrp)-1$				
[HL+byte]		3	2	–	$(HL+byte) \leftarrow (HL+byte)-1$				
ES: [HL+byte]		4	3	–	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)-1$				
Shift	SHR	A, cnt	2	1	–	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow 0) \times cnt$			x
	SHRW	AX, cnt	2	1	–	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow 0) \times cnt$			x
	SHL	A, cnt	2	1	–	$(CY \leftarrow A_7, A_m \leftarrow A_{m-1}, A_0 \leftarrow 0) \times cnt$			x
		B, cnt	2	1	–	$(CY \leftarrow B_7, B_m \leftarrow B_{m-1}, B_0 \leftarrow 0) \times cnt$			x
		C, cnt	2	1	–	$(CY \leftarrow C_7, C_m \leftarrow C_{m-1}, C_0 \leftarrow 0) \times cnt$			x
	SHLW	AX, cnt	2	1	–	$(CY \leftarrow AX_{15}, AX_m \leftarrow AX_{m-1}, AX_0 \leftarrow 0) \times cnt$			x
		BC, cnt	2	1	–	$(CY \leftarrow BC_{15}, BC_m \leftarrow BC_{m-1}, BC_0 \leftarrow 0) \times cnt$			x
	SAR	A, cnt	2	1	–	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow A_7) \times cnt$			x
SARW	AX, cnt	2	1	–	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow AX_{15}) \times cnt$			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 area is accessed, or when the data flash area 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 31-5. 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	–	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			x
	ROL	A, 1	2	1	–	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			x
	RORC	A, 1	2	1	–	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			x
	ROLC	A, 1	2	1	–	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			x
	ROLWC	AX,1	2	1	–	$(CY \leftarrow AX_{15}, AX_0 \leftarrow CY, AX_{m+1} \leftarrow AX_m) \times 1$			x
		BC,1	2	1	–	$(CY \leftarrow BC_{15}, BC_0 \leftarrow CY, BC_{m+1} \leftarrow BC_m) \times 1$			x
Bit manipulate	MOV1	CY, A.bit	2	1	–	$CY \leftarrow A.bit$			x
		A.bit, CY	2	1	–	$A.bit \leftarrow CY$			
		CY, PSW.bit	3	1	–	$CY \leftarrow PSW.bit$			x
		PSW.bit, CY	3	4	–	$PSW.bit \leftarrow CY$	x	x	
		CY, saddr.bit	3	1	–	$CY \leftarrow (saddr).bit$			x
		saddr.bit, CY	3	2	–	$(saddr).bit \leftarrow CY$			
		CY, sfr.bit	3	1	–	$CY \leftarrow sfr.bit$			x
		sfr.bit, CY	3	2	–	$sfr.bit \leftarrow CY$			
		CY,[HL].bit	2	1	4	$CY \leftarrow (HL).bit$			x
		[HL].bit, CY	2	2	–	$(HL).bit \leftarrow CY$			
	CY, ES:[HL].bit	3	2	5	$CY \leftarrow (ES, HL).bit$			x	
	ES:[HL].bit, CY	3	3	–	$(ES, HL).bit \leftarrow CY$				
	AND1	CY, A.bit	2	1	–	$CY \leftarrow CY \wedge A.bit$			x
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \wedge PSW.bit$			x
		CY, saddr.bit	3	1	–	$CY \leftarrow CY \wedge (saddr).bit$			x
		CY, sfr.bit	3	1	–	$CY \leftarrow CY \wedge sfr.bit$			x
		CY,[HL].bit	2	1	4	$CY \leftarrow CY \wedge (HL).bit$			x
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \wedge (ES, HL).bit$			x
	OR1	CY, A.bit	2	1	–	$CY \leftarrow CY \vee A.bit$			x
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \vee PSW.bit$			x
		CY, saddr.bit	3	1	–	$CY \leftarrow CY \vee (saddr).bit$			x
CY, sfr.bit		3	1	–	$CY \leftarrow CY \vee sfr.bit$			x	
CY, [HL].bit		2	1	4	$CY \leftarrow CY \vee (HL).bit$			x	
CY, ES:[HL].bit		3	2	5	$CY \leftarrow CY \vee (ES, HL).bit$			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 area is accessed, or when the data flash area 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 31-5. 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	–	$CY \leftarrow CY \nabla A.bit$			x
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \nabla PSW.bit$			x
		CY, saddr.bit	3	1	–	$CY \leftarrow CY \nabla (saddr).bit$			x
		CY, sfr.bit	3	1	–	$CY \leftarrow CY \nabla sfr.bit$			x
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \nabla (HL).bit$			x
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \nabla (ES, HL).bit$			x
	SET1	A.bit	2	1	–	$A.bit \leftarrow 1$			
		PSW.bit	3	4	–	$PSW.bit \leftarrow 1$	x	x	x
		!addr16.bit	4	2	–	$(addr16).bit \leftarrow 1$			
		ES:!addr16.bit	5	3	–	$(ES, addr16).bit \leftarrow 1$			
		saddr.bit	3	2	–	$(saddr).bit \leftarrow 1$			
		sfr.bit	3	2	–	$sfr.bit \leftarrow 1$			
		[HL].bit	2	2	–	$(HL).bit \leftarrow 1$			
		ES:[HL].bit	3	3	–	$(ES, HL).bit \leftarrow 1$			
	CLR1	A.bit	2	1	–	$A.bit \leftarrow 0$			
		PSW.bit	3	4	–	$PSW.bit \leftarrow 0$	x	x	x
		!addr16.bit	4	2	–	$(addr16).bit \leftarrow 0$			
		ES:!addr16.bit	5	3	–	$(ES, addr16).bit \leftarrow 0$			
		saddr.bit	3	2	–	$(saddr).bit \leftarrow 0$			
		sfr.bit	3	2	–	$sfr.bit \leftarrow 0$			
		[HL].bit	2	2	–	$(HL).bit \leftarrow 0$			
		ES:[HL].bit	3	3	–	$(ES, HL).bit \leftarrow 0$			
	SET1	CY	2	1	–	$CY \leftarrow 1$			1
	CLR1	CY	2	1	–	$CY \leftarrow 0$			0
	NOT1	CY	2	1	–	$CY \leftarrow \overline{CY}$			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 area is accessed, or when the data flash area 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 31-5. 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) <sub>s</sub> , (SP - 3) ← (PC+2) <sub>H</sub> , (SP - 4) ← (PC+2) <sub>L</sub> , PC ← CS, rp, SP ← SP - 4			
		\$!addr20	3	3	-	(SP - 2) ← (PC+3) <sub>s</sub> , (SP - 3) ← (PC+3) <sub>H</sub> , (SP - 4) ← (PC+3) <sub>L</sub> , PC ← PC+3+jdisp16, SP ← SP - 4			
		!addr16	3	3	-	(SP - 2) ← (PC+3) <sub>s</sub> , (SP - 3) ← (PC+3) <sub>H</sub> , (SP - 4) ← (PC+3) <sub>L</sub> , PC ← 0000, addr16, SP ← SP - 4			
		!!addr20	4	3	-	(SP - 2) ← (PC+4) <sub>s</sub> , (SP - 3) ← (PC+4) <sub>H</sub> , (SP - 4) ← (PC+4) <sub>L</sub> , PC ← addr20, SP ← SP - 4			
	CALLT	[addr5]	2	5	-	(SP - 2) ← (PC+2) <sub>s</sub> , (SP - 3) ← (PC+2) <sub>H</sub> , (SP - 4) ← (PC+2) <sub>L</sub> , PC <sub>s</sub> ← 0000, PC <sub>H</sub> ← (0000, addr5+1), PC <sub>L</sub> ← (0000, addr5), SP ← SP - 4			
	BRK	-	2	5	-	(SP - 1) ← PSW, (SP - 2) ← (PC+2) <sub>s</sub> , (SP - 3) ← (PC+2) <sub>H</sub> , (SP - 4) ← (PC+2) <sub>L</sub> , PC <sub>s</sub> ← 0000, PC <sub>H</sub> ← (0007FH), PC <sub>L</sub> ← (0007EH), SP ← SP - 4, IE ← 0			
	RET	-	1	6	-	PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP+1), PC <sub>s</sub> ← (SP+2), SP ← SP+4			
RETI	-	2	6	-	PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP+1), PC <sub>s</sub> ← (SP+2), PSW ← (SP+3), SP ← SP+4	R	R	R	
RETB	-	2	6	-	PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP+1), PC <sub>s</sub> ← (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 area is accessed, or when the data flash area 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 31-5. 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 <sub>H</sub> , (SP – 2) ← rp <sub>L</sub> , SP ← SP – 2			
	POP	PSW	2	3	–	PSW ← (SP+1), SP ← SP + 2	R	R	R
		rp	1	1	–	rp <sub>L</sub> ← (SP), rp <sub>H</sub> ← (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				
Un-conditional 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 <sup>Note3</sup>	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	\$addr20	2	2/4 <sup>Note3</sup>	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	\$addr20	2	2/4 <sup>Note3</sup>	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	\$addr20	2	2/4 <sup>Note3</sup>	–	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	\$addr20	3	2/4 <sup>Note3</sup>	–	PC ← PC + 3 + jdisp8 if (Z∨CY)=0			
	BNH	\$addr20	3	2/4 <sup>Note3</sup>	–	PC ← PC + 3 + jdisp8 if (Z∨CY)=1			
	BT	saddr.bit, \$addr20	4	3/5 <sup>Note3</sup>	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, \$addr20	4	3/5 <sup>Note3</sup>	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, \$addr20	3	3/5 <sup>Note3</sup>	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, \$addr20	4	3/5 <sup>Note3</sup>	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
[HL].bit, \$addr20		3	3/5 <sup>Note3</sup>	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1				
ES:[HL].bit, \$addr20	4	4/6 <sup>Note3</sup>	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1					

- Notes 1.** Number of CPU clocks (f<sub>CLK</sub>) 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<sub>CLK</sub>) when the code flash area is accessed, or when the data flash area 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 31-5. 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 <sup>Note3</sup>	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, \$addr20	4	3/5 <sup>Note3</sup>	–	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, \$addr20	3	3/5 <sup>Note3</sup>	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, \$addr20	4	3/5 <sup>Note3</sup>	–	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, \$addr20	3	3/5 <sup>Note3</sup>	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, \$addr20	4	4/6 <sup>Note3</sup>	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, \$addr20	4	3/5 <sup>Note3</sup>	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, \$addr20	4	3/5 <sup>Note3</sup>	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, \$addr20	3	3/5 <sup>Note3</sup>	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, \$addr20	4	3/5 <sup>Note3</sup>	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	x	x	x
		[HL].bit, \$addr20	3	3/5 <sup>Note3</sup>	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, \$addr20	4	4/6 <sup>Note3</sup>	–	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 <sup>Note4</sup>	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<sub>CLK</sub>) 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<sub>CLK</sub>) when the code flash area is accessed, or when the data flash area 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 the number of register banks (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 32 ELECTRICAL SPECIFICATIONS ( $T_A = -40$  to  $+85^\circ\text{C}$ )**

Target products A: Consumer applications;  $T_A = -40$  to  $+85^\circ\text{C}$

R5F10WLAAFA, R5F10WLCAFA, R5F10WLDafa,  
R5F10WLEAFA, R5F10WLFafa, R5F10WLGafa,  
R5F10WLAafb, R5F10WLCafb, R5F10WLDafb,  
R5F10WLEafb, R5F10WLFafb, R5F10WLGafb,  
R5F10WMAafa, R5F10WMcafa, R5F10WMDafa,  
R5F10WMEafa, R5F10WMfafa, R5F10WMGafa,  
R5F10WMAafb, R5F10WMcafb, R5F10WMDafb,  
R5F10WMEafb, R5F10WMfab, R5F10WMGafb

G: Industrial applications; when using  $T_A = -40$  to  $+105^\circ\text{C}$  specification products at  $T_A = -40$  to  $+85^\circ\text{C}$

R5F10WLAGFB, R5F10WLCGFB, R5F10WLDGFB,  
R5F10WLEGFB, R5F10WLFGB, R5F10WLGFB,  
R5F10WMAGFB, R5F10WMCGB, R5F10WMDGFB,  
R5F10WMEGFB, R5F10WMFGB, R5F10WMGFB

- 1. The RL78 microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.**
- 2. The pins mounted depend on the product. See 2.1 Port Function to 2.2.1 With functions for each product.**

### 32.1 Absolute Maximum Ratings

#### Absolute Maximum Ratings (1/3)

Parameter	Symbol	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 $V_{DD} + 0.3$ <sup>Note 1</sup>	V
Input voltage	$V_{I1}$	P00 to P07, P10 to P17, P20 to P27, P30 to P35, P40 to P47, P50 to P57, P60, P61, P70 to P77, P121 to P127, P130, P137	-0.3 to $V_{DD} + 0.3$ <sup>Note 2</sup>	V
	$V_{I2}$	P60 and P61 (N-ch open-drain)	-0.3 to +6.5	V
	$V_{I3}$	EXCLK, EXCLKS, $\overline{\text{RESET}}$	-0.3 to $V_{DD} + 0.3$ <sup>Note 2</sup>	V
Output voltage	$V_{O1}$	P00 to P07, P10 to P17, P20 to P27, P30 to P35, P40 to P47, P50 to P57, P60, P61, P70 to P77, P121 to P127, P130, P137	-0.3 to $V_{DD} + 0.3$ <sup>Note 2</sup>	V
Analog input voltage	$V_{AI1}$	ANI0, ANI1, ANI16 to ANI26	-0.3 to $V_{DD} + 0.3$ and -0.3 to $AV_{REF(+)} + 0.3$ <sup>Notes 2, 3</sup>	V

**Notes 1.** Connect the REGC pin to Vss via a capacitor (0.47 to 1  $\mu\text{F}$ ). 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.**  $AV_{REF (+)}$ : + side reference voltage of the A/D converter.

**3.** Vss: Reference voltage



**Absolute Maximum Ratings (2/3)**

Parameter	Symbol	Conditions	Ratings	Unit
LCD voltage	V <sub>L1</sub>	V <sub>L1</sub> voltage <sup>Note 1</sup>	-0.3 to +2.8 and -0.3 to V <sub>L4</sub> +0.3	V
	V <sub>L2</sub>	V <sub>L2</sub> voltage <sup>Note 1</sup>	-0.3 to V <sub>L4</sub> +0.3 <sup>Note 2</sup>	V
	V <sub>L3</sub>	V <sub>L3</sub> voltage <sup>Note 1</sup>	-0.3 to V <sub>L4</sub> +0.3 <sup>Note 2</sup>	V
	V <sub>L4</sub>	V <sub>L4</sub> voltage <sup>Note 1</sup>	-0.3 to +6.5	V
	V <sub>LCAP</sub>	CAPL, CAPH voltage <sup>Note 1</sup>	-0.3 to V <sub>L4</sub> +0.3 <sup>Note 2</sup>	V
	V <sub>OUT</sub>	COM0 to COM7 SEG0 to SEG50 output voltage	External resistance division method	-0.3 to V <sub>DD</sub> +0.3 <sup>Note 2</sup>
		Capacitor split method	-0.3 to V <sub>DD</sub> +0.3 <sup>Note 2</sup>	V
		Internal voltage boosting method	-0.3 to V <sub>L4</sub> +0.3 <sup>Note 2</sup>	V

**Notes 1.** This value only indicates the absolute maximum ratings when applying voltage to the V<sub>L1</sub>, V<sub>L2</sub>, V<sub>L3</sub>, and V<sub>L4</sub> pins; it does not mean that applying voltage to these pins is recommended. When using the internal voltage boosting method or capacitance split method, connect these pins to V<sub>SS</sub> via a capacitor (0.47 μF ± 30%) and connect a capacitor (0.47 μF ± 30%) between the CAPL and CAPH pins.

**2.** Must be 6.5 V or lower.

**Caution** Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

**Remark** V<sub>SS</sub>: Reference voltage

**Absolute Maximum Ratings (3/3)**

Parameter	Symbol	Conditions		Ratings	Unit
<R> Output current, high	$I_{OH1}$	Per pin	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P60, P61, P70 to P77, P125 to P127, P130	-40	mA
<R>		Total of all pins -170 mA	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P60, P61, P70 to P77, P125 to P127, P130	-170	mA
<R>	$I_{OH2}$	Per pin	P20, P21	-0.5	mA
<R>		Total of all pins		-1	mA
<R> Output current, low	$I_{OL1}$	Per pin	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P60, P61, P70 to P77, P125 to P127, P130	40	mA
<R>		Total of all pins 170 mA	P40 to P47, P130	70	mA
<R>			P00 to P07, P10 to P17, P22 to P27, P30 to P35, P50 to P57, P60, P61, P70 to P77, P125 to P127	100	mA
<R>	$I_{OL2}$	Per pin	P20, P21	1	mA
<R>		Total of all pins		2	mA
Operating ambient temperature	$T_A$	In normal operation mode		-40 to +85	$^\circ\text{C}$
		In flash memory programming mode			
Storage temperature	$T_{stg}$			-65 to +150	$^\circ\text{C}$

**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.

## 32.2 Oscillator Characteristics

### 32.2.1 X1 and XT1 oscillator characteristics

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency ( $f_x$ ) <sup>Note</sup>	Ceramic resonator/ crystal resonator	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	1.0		20.0	MHz
		$2.4\text{ V} \leq V_{DD} < 2.7\text{ V}$	1.0		16.0	
		$1.8\text{ V} \leq V_{DD} < 2.4\text{ V}$	1.0		8.0	
		$1.6\text{ V} \leq V_{DD} < 1.8\text{ V}$	1.0		4.0	
XT1 clock oscillation frequency ( $f_{XT}$ ) <sup>Note</sup>	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, see 5.4 System Clock Oscillator.

**32.2.2 On-chip oscillator characteristics****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency <sup>Notes 1, 2</sup>	$f_{IH}$			1		24	MHz
High-speed on-chip oscillator clock frequency accuracy		-20 to $+85^\circ\text{C}$	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	-1.0		+1.0	%
			$1.6\text{ V} \leq V_{DD} < 1.8\text{ V}$	-5.0		+5.0	%
		-40 to $-20^\circ\text{C}$	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	-1.5		+1.5	%
			$1.6\text{ V} \leq V_{DD} < 1.8\text{ V}$	-5.5		+5.5	%
Low-speed on-chip oscillator clock frequency	$f_{IL}$				15		kHz
Low-speed on-chip oscillator clock frequency accuracy				-15		+15	%

- Notes 1.** The high-speed on-chip oscillator frequency is selected by bits 0 to 4 of the option byte (000C2H/010C2H) and bits 0 to 2 of the HOCODIV register.
- 2.** This indicates the oscillator characteristics only. Refer to **AC Characteristics** for the instruction execution time.

### 32.3 DC Characteristics

#### 32.3.1 Pin characteristics

(T<sub>A</sub> = -40 to +85°C, 1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

<R>

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Output current, high <sup>Note 1</sup>	I <sub>OH1</sub>	Per pin for P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130	1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V			-10.0 <sup>Note 2</sup>	mA
		Total of P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130 (When duty = 70% <sup>Note 3</sup> )	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V			-90.0	mA
			2.7 V ≤ V <sub>DD</sub> < 4.0 V			-15.0	mA
			1.8 V ≤ V <sub>DD</sub> < 2.7 V			-7.0	mA
	I <sub>OH2</sub>	Per pin for P20 and P21	1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V			-0.1 <sup>Note 2</sup>	mA
			Total of all pins (When duty = 70% <sup>Note 3</sup> )	1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V			-0.2

- Notes**
1. Value of the current at which the device operation is guaranteed even if the current flows from the V<sub>DD</sub> pin to an output pin
  2. Do not exceed the total current value.
  3. Output current value 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 factor from 70% to n%).

- Total output current of pins = (I<sub>OH</sub> × 0.7)/(n × 0.01)

<Example> Where n = 80% and I<sub>OH</sub> = -90.0 mA

$$\text{Total output current of pins} = (-90.0 \times 0.7)/(80 \times 0.01) \cong -78.75 \text{ mA}$$

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, P04 to P07, P16, P17, P35, P42 to P44, P46, P47, P53 to P56, and P130 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<sub>A</sub> = -40 to +85°C, 1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Output current, low <sup>Note 1</sup>	I <sub>OL1</sub>	Per pin for P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130			20.0 <sup>Note 2</sup>	mA	
		Per pin for P60 and P61			15.0 <sup>Note 2</sup>	mA	
		Total of P40 to P47, P130 (When duty = 70% <sup>Note 3</sup> )	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V			70.0	mA
			2.7 V ≤ V <sub>DD</sub> < 4.0 V			15.0	mA
			1.8 V ≤ V <sub>DD</sub> < 2.7 V			9.0	mA
			1.6 V ≤ V <sub>DD</sub> < 1.8 V			4.5	mA
		Total of P00 to P07, P10 to P17, P22 to P27, P30 to P35, P50 to P57, P70 to P77, P125 to P127 (When duty = 70% <sup>Note 3</sup> )	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V			90.0	mA
			2.7 V ≤ V <sub>DD</sub> < 4.0 V			35.0	mA
			1.8 V ≤ V <sub>DD</sub> < 2.7 V			20.0	mA
			1.6 V ≤ V <sub>DD</sub> < 1.8 V			10.0	mA
	Total of all pins (When duty = 70% <sup>Note 3</sup> )				160.0	mA	
	I <sub>OL2</sub>	Per pin for P20 and P21				0.4 <sup>Note 2</sup>	mA
		Total of all pins (When duty = 70% <sup>Note 3</sup> )	1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V			0.8	mA

<R>

- Notes**
- Value of the current at which the device operation is guaranteed even if the current flows from an output pin to the V<sub>SS</sub> pin
  - Do not exceed the total current value.
  - Output current value 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 factor from 70% to n%).
    - Total output current of pins = (I<sub>OL</sub> × 0.7)/(n × 0.01)  
 <Example> Where n = 80% and I<sub>OL</sub> = 70.0 mA  
 Total output current of pins = (70.0 × 0.7)/(80 × 0.01) ≅ 61.25 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  $+85^\circ\text{C}$ ,  $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input voltage, high	$V_{IH1}$	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130, P137	Normal input buffer	$0.8V_{DD}$		$V_{DD}$	V
	$V_{IH2}$	P03, P05, P06, P16, P17, P34, P43, P44, P46, P47, P53, P55	TTL input buffer $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.2		$V_{DD}$	V
			TTL input buffer $3.3\text{ V} \leq V_{DD} < 4.0\text{ V}$	2.0		$V_{DD}$	V
			TTL input buffer $1.6\text{ V} \leq V_{DD} < 3.3\text{ V}$	1.5		$V_{DD}$	V
	$V_{IH3}$	P20, P21		$0.7V_{DD}$		$V_{DD}$	V
	$V_{IH4}$	P60, P61		$0.7V_{DD}$		6.0	V
	$V_{IH5}$	P121 to P124, P137, EXCLK, EXCLKS, RESET		$0.8V_{DD}$		$V_{DD}$	V
Input voltage, low	$V_{IL1}$	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130, P137	Normal input buffer	0		$0.2V_{DD}$	V
	$V_{IL2}$	P03, P05, P06, P16, P17, P34, P43, P44, P46, P47, P53, P55	TTL input buffer $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0		0.8	V
			TTL input buffer $3.3\text{ V} \leq V_{DD} < 4.0\text{ V}$	0		0.5	V
			TTL input buffer $1.6\text{ V} \leq V_{DD} < 3.3\text{ V}$	0		0.32	V
	$V_{IL3}$	P20, P21		0		$0.3V_{DD}$	V
	$V_{IL4}$	P60, P61		0		$0.3V_{DD}$	V
	$V_{IL5}$	P121 to P124, P137, EXCLK, EXCLKS, RESET		0		$0.2V_{DD}$	V

**Caution** The maximum value of  $V_{IH}$  of pins P00, P04 to P07, P16, P17, P35, P42 to P44, P46, P47, P53 to P56, and P130 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  $+85^\circ\text{C}$ ,  $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Output voltage, high	V <sub>OH1</sub>	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OH1} = -10.0\text{ mA}$	$V_{DD} - 1.5$			V
			$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OH1} = -3.0\text{ mA}$	$V_{DD} - 0.7$			V
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OH1} = -2.0\text{ mA}$	$V_{DD} - 0.6$			V
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OH1} = -1.5\text{ mA}$	$V_{DD} - 0.5$			V
			$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OH1} = -1.0\text{ mA}$	$V_{DD} - 0.5$			V
	V <sub>OH2</sub>	P20 and P21	$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OH2} = -100\ \mu\text{A}$	$V_{DD} - 0.5$			V
Output voltage, low	V <sub>OL1</sub>	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 20\text{ mA}$			1.3	V
			$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 8.5\text{ mA}$			0.7	V
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 3.0\text{ mA}$			0.6	V
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 1.5\text{ mA}$			0.4	V
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 0.6\text{ mA}$			0.4	V
			$1.6\text{ V} \leq V_{DD} < 1.8\text{ V}$ , $I_{OL1} = 0.3\text{ mA}$			0.4	V
	V <sub>OL2</sub>	P20 and P21	$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL2} = 400\ \mu\text{A}$			0.4	V
	V <sub>OL3</sub>	P60 and P61	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL3} = 15.0\text{ mA}$			2.0	V
			$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL3} = 5.0\text{ mA}$			0.4	V
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL3} = 3.0\text{ mA}$			0.4	V
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL3} = 2.0\text{ mA}$			0.4	V
			$1.6\text{ V} \leq V_{DD} < 1.8\text{ V}$ , $I_{OL3} = 1.0\text{ mA}$			0.4	V

**Caution** P00, P04 to P07, P16, P17, P35, P42 to P44, P46, P47, P53 to P56, and P130 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  $+85^\circ\text{C}$ ,  $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit		
Input leakage current, high	I <sub>LIH1</sub>	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130, P137	$V_I = V_{DD}$			1	$\mu\text{A}$	
	I <sub>LIH2</sub>	P20 and P21, $\overline{\text{RESET}}$	$V_I = V_{DD}$			1	$\mu\text{A}$	
	I <sub>LIH3</sub>	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{DD}$	In input port mode and when external clock is input			1	$\mu\text{A}$
				Resonator connected			10	$\mu\text{A}$
Input leakage current, low	I <sub>LIL1</sub>	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130, P137	$V_I = V_{SS}$			-1	$\mu\text{A}$	
	I <sub>LIL2</sub>	P20 and P21, $\overline{\text{RESET}}$	$V_I = V_{SS}$			-1	$\mu\text{A}$	
	I <sub>LIL3</sub>	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{SS}$	In input port mode and when external clock is input			-1	$\mu\text{A}$
				Resonator connected			-10	$\mu\text{A}$
On-chip pull-up resistance	R <sub>U1</sub>	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P45 to P47, P50 to P57, P70 to P77, P125 to P127, P130	$V_I = V_{SS}$	$2.4\text{ V} \leq V_{DD} < 5.5\text{ V}$	10	20	100	$\text{k}\Omega$
				$1.6\text{ V} \leq V_{DD} < 2.4\text{ V}$	10	30	100	$\text{k}\Omega$
	R <sub>U2</sub>	P40 to P44	$V_I = V_{SS}$	10	20	100	$\text{k}\Omega$	

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

## 32.3.2 Supply current characteristics

(T<sub>A</sub> = -40 to +85°C, 1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

(1/2)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit	
Supply current <sup>Note 1</sup>	I <sub>DD1</sub>	Operating mode	HS (high-speed main) mode <sup>Note 5</sup>	f <sub>HOCO</sub> = 48 MHz <sup>Note 3</sup> , f <sub>IH</sub> = 24 MHz <sup>Note 3</sup>	Basic operation	V <sub>DD</sub> = 5.0 V		2.0		mA
						V <sub>DD</sub> = 3.0 V		2.0		mA
				Normal operation	V <sub>DD</sub> = 5.0 V		3.8	6.5	mA	
					V <sub>DD</sub> = 3.0 V		3.8	6.5	mA	
				Basic operation	V <sub>DD</sub> = 5.0 V		1.7		mA	
					V <sub>DD</sub> = 3.0 V		1.7		mA	
			Normal operation	V <sub>DD</sub> = 5.0 V		3.6	6.1	mA		
				V <sub>DD</sub> = 3.0 V		3.6	6.1	mA		
			Normal operation	V <sub>DD</sub> = 5.0 V		2.7	4.7	mA		
				V <sub>DD</sub> = 3.0 V		2.7	4.7	mA		
			LS (low-speed main) mode <sup>Note 5</sup>	f <sub>HOCO</sub> = 8 MHz <sup>Note 3</sup> , f <sub>IH</sub> = 8 MHz <sup>Note 3</sup>	Normal operation	V <sub>DD</sub> = 3.0 V		1.2	2.1	mA
						V <sub>DD</sub> = 2.0 V		1.2	2.1	mA
		LV (low-voltage main) mode <sup>Note 5</sup>	f <sub>HOCO</sub> = 4 MHz <sup>Note 3</sup> , f <sub>IH</sub> = 4 MHz <sup>Note 3</sup>	Normal operation	V <sub>DD</sub> = 3.0 V		1.2	1.8	mA	
					V <sub>DD</sub> = 2.0 V		1.2	1.8	mA	
		HS (high-speed main) mode <sup>Note 5</sup>	f <sub>MX</sub> = 20 MHz <sup>Note 2</sup> , V <sub>DD</sub> = 5.0 V	Normal operation	Square wave input		3.0	5.1	mA	
					Resonator connection		3.2	5.2	mA	
				Normal operation	Square wave input		2.9	5.1	mA	
					Resonator connection		3.2	5.2	mA	
				Normal operation	Square wave input		2.5	4.4	mA	
					Resonator connection		2.7	4.5	mA	
			Normal operation	Square wave input		2.5	4.4	mA		
				Resonator connection		2.7	4.5	mA		
			Normal operation	Square wave input		1.9	3.0	mA		
				Resonator connection		1.9	3.0	mA		
			Normal operation	Square wave input		1.9	3.0	mA		
				Resonator connection		1.9	3.0	mA		
		LS (low-speed main) mode <sup>Note 5</sup>	f <sub>MX</sub> = 8 MHz <sup>Note 2</sup> , V <sub>DD</sub> = 3.0 V	Normal operation	Square wave input		1.1	2.0	mA	
					Resonator connection		1.1	2.0	mA	
			Normal operation	Square wave input		1.1	2.0	mA		
				Resonator connection		1.1	2.0	mA		
		Subsystem clock operation	f <sub>SUB</sub> = 32.768 kHz <sup>Note 4</sup> , T <sub>A</sub> = -40°C	Normal operation	Square wave input		4.0	5.4	μA	
					Resonator connection		4.3	5.4	μA	
Normal operation	Square wave input			4.0	5.4	μA				
	Resonator connection			4.3	5.4	μA				
Normal operation	Square wave input			4.1	7.1	μA				
	Resonator connection			4.4	7.1	μA				
Normal operation	Square wave input			4.3	8.7	μA				
	Resonator connection			4.7	8.7	μA				
Normal operation	Square wave input			4.7	12.0	μA				
	Resonator connection			5.2	12.0	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 values below the MAX. column include the peripheral operation current. However, not including the current flowing into the LCD controller/driver, A/D converter, LVD circuit, comparator, I/O port, on-chip pull-up/pull-down resistors, and the current flowing during data flash rewrite.
  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 setting ultra-low power consumption oscillation ( $AMPHS1 = 1$ ). The current flowing into the LCD controller/driver, 16-bit timer KB20, real-time clock 2, 12-bit interval timer, and watchdog timer is not included.
  5. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.  
HS (high-speed main) mode:  $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz to }24\text{ MHz}$   
 $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz to }16\text{ MHz}$   
LS (low-speed main) mode:  $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz to }8\text{ MHz}$   
LV (low-voltage main) mode:  $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz to }4\text{ MHz}$

- Remarks**
1.  $f_{MX}$ : High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)
  2.  $f_{HOCO}$ : High-speed on-chip oscillator clock frequency (48 MHz max.)
  3.  $f_{IH}$ : High-speed on-chip oscillator clock frequency (24 MHz max.)
  4.  $f_{SUB}$ : Subsystem clock frequency (XT1 clock oscillation frequency)
  5. Except subsystem clock operation, temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$

(T<sub>A</sub> = -40 to +85°C, 1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

(2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit			
Supply current <sup>Note 1</sup>	I <sub>DD2</sub> <sup>Note 2</sup>	HALT mode	HS (high-speed main) mode <sup>Note 7</sup>	f <sub>HOCO</sub> = 48 MHz <sup>Note 4</sup> , f <sub>IH</sub> = 24 MHz <sup>Note 4</sup>	V <sub>DD</sub> = 5.0 V	0.71	1.95	mA		
					V <sub>DD</sub> = 3.0 V	0.71	1.95			
				f <sub>HOCO</sub> = 24 MHz <sup>Note 4</sup> , f <sub>IH</sub> = 24 MHz <sup>Note 4</sup>	V <sub>DD</sub> = 5.0 V	0.49	1.64	mA		
					V <sub>DD</sub> = 3.0 V	0.49	1.64			
				f <sub>HOCO</sub> = 16 MHz <sup>Note 4</sup> , f <sub>IH</sub> = 16 MHz <sup>Note 4</sup>	V <sub>DD</sub> = 5.0 V	0.43	1.11	mA		
					V <sub>DD</sub> = 3.0 V	0.43	1.11			
			LS (low-speed main) mode <sup>Note 7</sup>	f <sub>HOCO</sub> = 8 MHz <sup>Note 4</sup> , f <sub>IH</sub> = 8 MHz <sup>Note 4</sup>	V <sub>DD</sub> = 3.0 V	280	770	μA		
					V <sub>DD</sub> = 2.0 V	280	770			
			LV (low-voltage main) mode <sup>Note 7</sup>	f <sub>HOCO</sub> = 4 MHz <sup>Note 4</sup> , f <sub>IH</sub> = 4 MHz <sup>Note 4</sup>	V <sub>DD</sub> = 3.0 V	430	700	μA		
					V <sub>DD</sub> = 2.0 V	430	700			
			HS (high-speed main) mode <sup>Note 7</sup>			f <sub>MX</sub> = 20 MHz <sup>Note 3</sup> , V <sub>DD</sub> = 5.0 V	Square wave input	0.31	1.42	mA
							Resonator connection	0.48	1.42	
						f <sub>MX</sub> = 20 MHz <sup>Note 3</sup> , V <sub>DD</sub> = 3.0 V	Square wave input	0.29	1.42	mA
							Resonator connection	0.48	1.42	
						f <sub>MX</sub> = 16 MHz <sup>Note 3</sup> , V <sub>DD</sub> = 5.0 V	Square wave input	0.26	0.86	mA
							Resonator connection	0.45	1.15	
						f <sub>MX</sub> = 16 MHz <sup>Note 3</sup> , V <sub>DD</sub> = 3.0 V	Square wave input	0.25	0.86	mA
							Resonator connection	0.44	1.15	
	f <sub>MX</sub> = 10 MHz <sup>Note 3</sup> , V <sub>DD</sub> = 5.0 V	Square wave input				0.20	0.63	mA		
		Resonator connection				0.28	0.71			
	f <sub>MX</sub> = 10 MHz <sup>Note 3</sup> , V <sub>DD</sub> = 3.0 V	Square wave input				0.19	0.63	mA		
		Resonator connection				0.28	0.71			
	LS (low-speed main) mode <sup>Note 7</sup>			f <sub>MX</sub> = 8 MHz <sup>Note 3</sup> , V <sub>DD</sub> = 3.0 V	Square wave input	100	560	μA		
					Resonator connection	160	560			
				f <sub>MX</sub> = 8 MHz <sup>Note 3</sup> , V <sub>DD</sub> = 2.0 V	Square wave input	100	560	μA		
					Resonator connection	160	560			
	Subsystem clock operation			f <sub>SUB</sub> = 32.768 kHz <sup>Note 5</sup> , T <sub>A</sub> = -40°C	Square wave input	0.34	0.62	μA		
Resonator connection					0.51	0.80				
f <sub>SUB</sub> = 32.768 kHz <sup>Note 5</sup> , T <sub>A</sub> = +25°C				Square wave input	0.38	0.62	μA			
				Resonator connection	0.57	0.80				
f <sub>SUB</sub> = 32.768 kHz <sup>Note 5</sup> , T <sub>A</sub> = +50°C				Square wave input	0.46	2.30	μA			
				Resonator connection	0.67	2.49				
f <sub>SUB</sub> = 32.768 kHz <sup>Note 5</sup> , T <sub>A</sub> = +70°C				Square wave input	0.65	4.03	μA			
				Resonator connection	0.91	4.22				
f <sub>SUB</sub> = 32.768 kHz <sup>Note 5</sup> , T <sub>A</sub> = +85°C				Square wave input	1.00	8.04	μA			
				Resonator connection	1.31	8.23				
I <sub>DD3</sub> <sup>Note 6</sup>	STOP mode <sup>Note 8</sup>		T <sub>A</sub> = -40°C		0.18	0.52	μA			
				T <sub>A</sub> = +25°C	0.24	0.52				
				T <sub>A</sub> = +50°C	0.33	2.21				
				T <sub>A</sub> = +70°C	0.53	3.94				
				T <sub>A</sub> = +85°C	0.93	7.95				

(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 values below the MAX. column include the peripheral operation current. However, not including the current flowing into the LCD controller/driver, A/D converter, LVD circuit, comparator, I/O port, on-chip pull-up/pull-down resistors, and the current flowing during data flash rewrite.
  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  $\text{RTCLPC} = 1$  and setting ultra-low current consumption ( $\text{AMPHS1} = 1$ ). The current flowing into the real-time clock 2 is included. However, not including the current flowing into the clock output/buzzer output, 12-bit interval timer, and watchdog timer.
  6. Not including the current flowing into the real-time clock 2, clock output/buzzer output, 12-bit interval timer, and watchdog timer.
  7. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.  
HS (high-speed main) mode:  $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz to }24\text{ MHz}$   
 $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz to }16\text{ MHz}$   
LS (low-speed main) mode:  $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz to }8\text{ MHz}$   
LV (low-voltage main) mode:  $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz to }4\text{ MHz}$
  8. Regarding the value for current to operate the subsystem clock in STOP mode, refer to that in HALT mode.

- Remarks**
1.  $f_{MX}$ : High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)
  2.  $f_{HOCO}$ : High-speed on-chip oscillator clock frequency (48 MHz max.)
  3.  $f_{IH}$ : High-speed on-chip oscillator clock frequency (24 MHz max.)
  4.  $f_{SUB}$ : Subsystem clock frequency (XT1 clock oscillation frequency)
  5. Except subsystem clock operation and STOP mode, temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$

(T<sub>A</sub> = -40 to +85°C, 1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I <sub>FIL</sub> <sup>Note 1</sup>						0.20		μA
RTC2 operating current	I <sub>RTC</sub> <sup>Notes 1, 2, 3</sup>	f <sub>SUB</sub> = 32.768 kHz					0.02		μA
12-bit interval timer operating current	I <sub>TMKA</sub> <sup>Notes 1, 2, 4</sup>						0.04		μA
Watchdog timer operating current	I <sub>WDT</sub> <sup>Notes 1, 2, 5</sup>	f <sub>IL</sub> = 15 kHz					0.22		μA
A/D converter operating current	I <sub>ADC</sub> <sup>Notes 1, 6</sup>	When conversion at maximum speed	Normal mode, AV <sub>REFP</sub> = V <sub>DD</sub> = 5.0 V				1.3	1.7	mA
			Low voltage mode, AV <sub>REFP</sub> = V <sub>DD</sub> = 3.0 V				0.5	0.7	mA
A/D converter reference voltage current	I <sub>ADREF</sub> <sup>Note 1</sup>						75.0		μA
Temperature sensor operating current	I <sub>TMPS</sub> <sup>Note 1</sup>						75.0		μA
LVD operating current	I <sub>LVD</sub> <sup>Notes 1, 7</sup>						0.08		μA
Comparator operating current	I <sub>COMP</sub> <sup>Notes 1, 11</sup>	V <sub>DD</sub> = 5.0 V, Regulator output voltage = 2.1 V	Window mode				12.5		μA
			Comparator high-speed mode				6.5		μA
			Comparator low-speed mode				1.7		μA
		V <sub>DD</sub> = 5.0 V, Regulator output voltage = 1.8 V	Window mode				8.0		μA
			Comparator high-speed mode				4.0		μA
			Comparator low-speed mode				1.3		μA
Self-programming operating current	I <sub>FSP</sub> <sup>Notes 1, 9</sup>						2.00	12.20	mA
BGO operating current	I <sub>BGO</sub> <sup>Notes 1, 8</sup>						2.00	12.20	mA
SNOOZE operating current	I <sub>SNOZ</sub> <sup>Note 1</sup>	ADC operation	While the mode is shifting <sup>Note 10</sup>				0.50	0.60	mA
			During A/D conversion, in low voltage mode, AV <sub>REFP</sub> = V <sub>DD</sub> = 3.0 V				1.20	1.44	mA
		CSI/UART operation				0.70	0.84	mA	
LCD operating current	I <sub>LCD1</sub> <sup>Notes 1, 12, 13</sup>	External resistance division method	f <sub>LCD</sub> = f <sub>SUB</sub>	1/3 bias, four time slices	V <sub>DD</sub> = 5.0 V, V <sub>L4</sub> = 5.0 V		0.04	0.20	μA
			LCD clock = 128 Hz						
	I <sub>LCD2</sub> <sup>Note 1, 12</sup>	Internal voltage boosting method	f <sub>LCD</sub> = f <sub>SUB</sub>	1/3 bias, four time slices	V <sub>DD</sub> = 3.0 V, V <sub>L4</sub> = 3.0 V (V <sub>LCD</sub> = 04H)		0.85	2.20	μA
			LCD clock = 128 Hz				V <sub>DD</sub> = 5.0 V, V <sub>L4</sub> = 5.1 V (V <sub>LCD</sub> = 12H)		1.55
I <sub>LCD3</sub> <sup>Note 1, 12</sup>	Capacitor split method	f <sub>LCD</sub> = f <sub>SUB</sub>	1/3 bias, four time slices	V <sub>DD</sub> = 3.0 V, V <sub>L4</sub> = 3.0 V		0.20	0.50	μA	
		LCD clock = 128 Hz							

(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 2 (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The value of the current for the RL78 microcontrollers is the sum of the values of either  $I_{DD1}$  or  $I_{DD2}$ , and  $I_{RTC}$ , when the real-time clock 2 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 real-time clock 2.
  4. Current flowing only to the 12-bit interval timer (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The value of the current for the RL78 microcontrollers is the sum of the values of either  $I_{DD1}$  or  $I_{DD2}$ , and  $I_{TMKA}$ , 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 current value of the RL78 microcontrollers is the sum of  $I_{DD1}$ ,  $I_{DD2}$  or  $I_{DD3}$  and  $I_{WDT}$  when the watchdog timer operates.
  6. Current flowing only to the A/D converter. The current value 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 microcontrollers is the sum of  $I_{DD1}$ ,  $I_{DD2}$  or  $I_{DD3}$  and  $I_{LVD}$  when the LVD circuit operates.
  8. Current flowing only during data flash rewrite.
  9. Current flowing only during self programming.
  10. For shift time to the SNOOZE mode, see **21.3.3 SNOOZE mode**.
  11. Current flowing only to the comparator circuit. The current value of the RL78 microcontrollers is the sum of  $I_{DD1}$ ,  $I_{DD2}$  or  $I_{DD3}$  and  $I_{CMP}$  when the comparator circuit operates.
  12. Current flowing only to the LCD controller/driver. The value of the current for the RL78 microcontrollers is the sum of the supply current ( $I_{DD1}$  or  $I_{DD2}$ ) and LCD operating current ( $I_{LCD1}$ ,  $I_{LCD2}$ , or  $I_{LCD3}$ ), when the LCD controller/driver operates in operation mode or HALT mode. However, not including the current flowing into the LCD panel. Conditions of the TYP. value and MAX. value are as follows.
    - Setting 20 pins as the segment function and blinking all
    - Selecting  $f_{SUB}$  for system clock when LCD clock = 128 Hz ( $LCDC0 = 07H$ )
    - Setting four time slices and 1/3 bias
  13. Not including the current flowing into the external division resistor when using the external resistance division method.

- 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. The temperature condition for the TYP. value is  $T_A = 25^\circ\text{C}$ .

## 32.4 AC Characteristics

(T<sub>A</sub> = -40 to +85°C, 1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	
Instruction cycle (minimum instruction execution time)	T <sub>CY</sub>	Main system clock (f <sub>MAIN</sub> ) operation	HS (high-speed main) mode	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.0417	1	μs	
				2.4 V ≤ V <sub>DD</sub> < 2.7 V	0.0625	1	μs	
			LS (low-speed main) mode	1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.125	1	μs	
				1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.25	1	μs	
		Subsystem clock (f <sub>SUB</sub> ) operation <sup>Note</sup>		1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V	28.5	30.5	31.3	μs
		In the self programming mode	HS (high-speed main) mode	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.0417	1	μs	
				2.4 V ≤ V <sub>DD</sub> < 2.7 V	0.0625	1	μs	
			LS (low-speed main) mode	1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.125	1	μs	
1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.25			1	μs			
External system clock frequency	f <sub>EX</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V		1.0		20.0	MHz	
		2.4 V ≤ V <sub>DD</sub> < 2.7 V		1.0		16.0	MHz	
		1.8 V ≤ V <sub>DD</sub> < 2.4 V		1.0		8.0	MHz	
		1.6 V ≤ V <sub>DD</sub> < 1.8 V		1.0		4.0	MHz	
	f <sub>EXS</sub>			32		35	kHz	
External system clock input high-level width, low-level width	t <sub>EXH</sub> , t <sub>EXL</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V		24			ns	
		2.4 V ≤ V <sub>DD</sub> < 2.7 V		30			ns	
		1.8 V ≤ V <sub>DD</sub> < 2.4 V		60			ns	
		1.6 V ≤ V <sub>DD</sub> < 1.8 V		120			ns	
	t <sub>EXHS</sub> , t <sub>EXLS</sub>			13.7			μs	
TI00 to TI07 input high-level width, low-level width	t <sub>TIH</sub> , t <sub>TIL</sub>			1/f <sub>MCK</sub> +10			ns	
TO00 to TO07, TKBO00, TKBO01-0 to TKBO01-2 output frequency	f <sub>TO</sub>	HS (high-speed main) mode	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V			12	MHz	
			2.7 V ≤ V <sub>DD</sub> < 4.0 V			8	MHz	
			2.4 V ≤ V <sub>DD</sub> < 2.7 V			4	MHz	
		LV (low-voltage main) mode	1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V				2	MHz
		LS (low-speed main) mode	1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V				4	MHz
PCLBUZ0, PCLBUZ1 output frequency	f <sub>PCL</sub>	HS (high-speed main) mode	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V			16	MHz	
			2.7 V ≤ V <sub>DD</sub> < 4.0 V			8	MHz	
			2.4 V ≤ V <sub>DD</sub> < 2.7 V			4	MHz	
		LV (low-voltage main) mode	1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V				4	MHz
			1.6 V ≤ V <sub>DD</sub> < 1.8 V				2	MHz
		LS (low-speed main) mode	1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V				4	MHz
Interrupt input high-level width, low-level width	t <sub>INTH</sub> , t <sub>INTL</sub>	INTP0 to INTP7		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	1		μs	
		Key interrupt input high-level width, low-level width	KR0 to KR7	1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V	250		ns	
1.6 V ≤ V <sub>DD</sub> < 1.8 V	1				μs			
IH-PWM output restart input high-level width	t <sub>IHR</sub>	INTP0 to INTP7			2		f <sub>CLK</sub>	
TMKB2 forced output stop input high-level width	t <sub>IHR</sub>	INTP0 to INTP2			2		f <sub>CLK</sub>	
RESET low-level width	t <sub>RSL</sub>				10		μs	

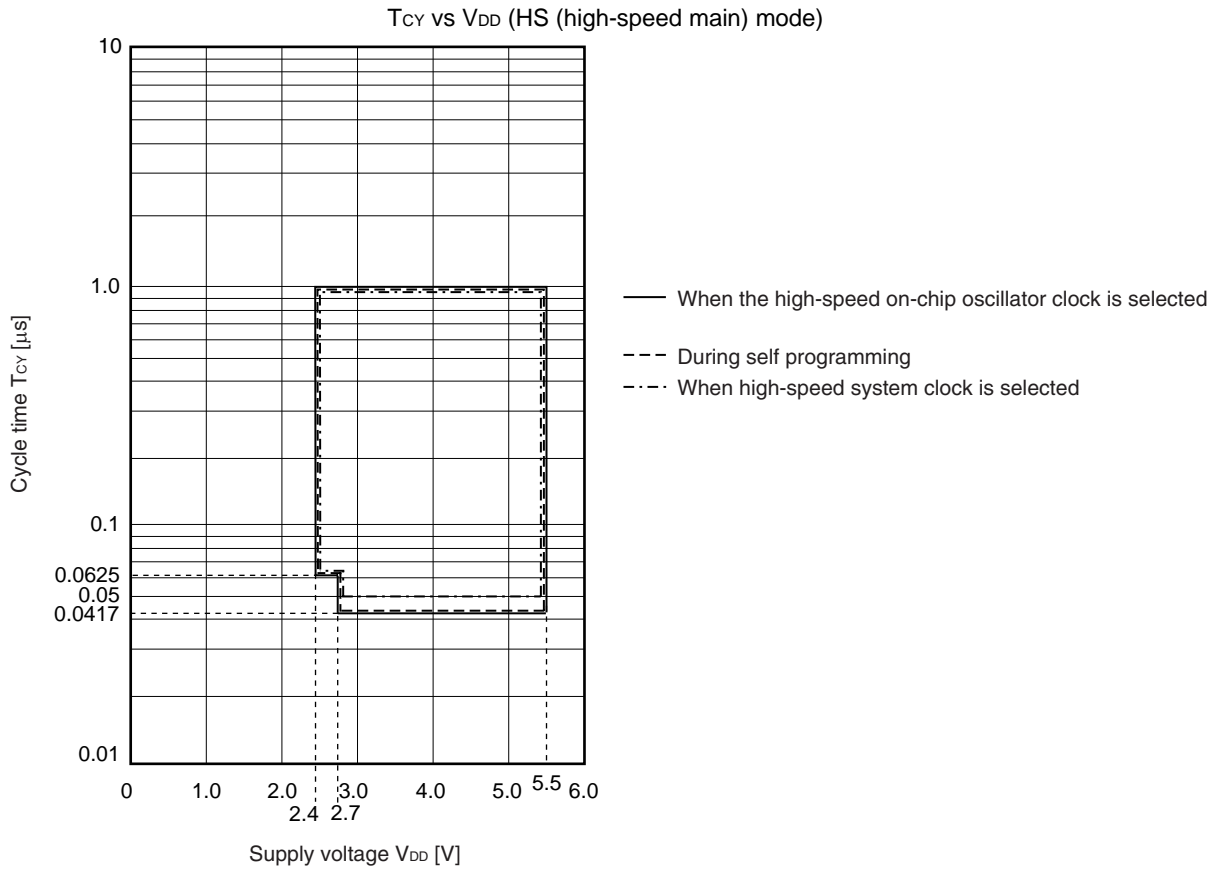
(Note and Remark are listed on the next page.)



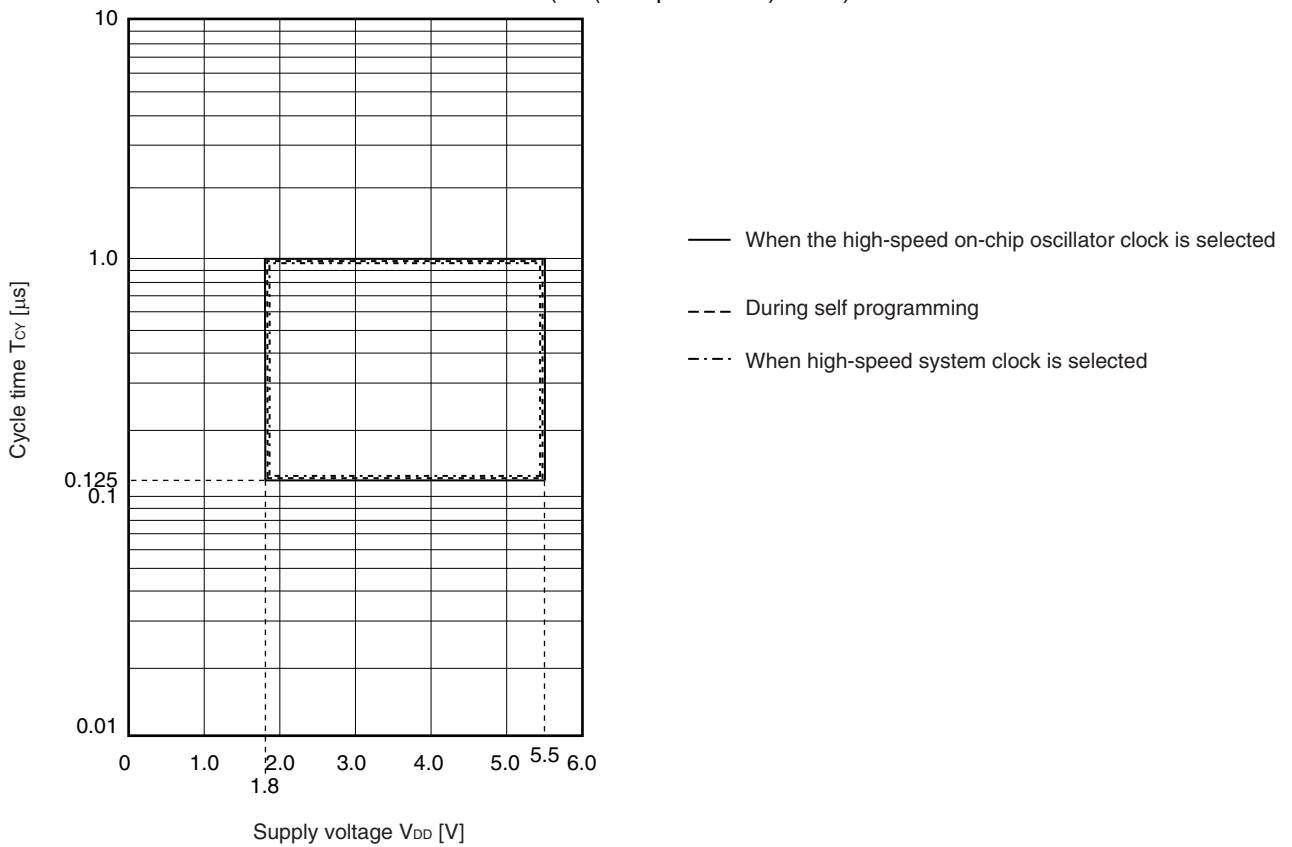
**Note** Operation is not possible if  $1.6\text{ V} \leq V_{DD} < 1.8\text{ V}$  in LV (low-voltage main) mode while the system is operating on the subsystem clock.

**Remark**  $f_{MCK}$ : Timer array unit operation clock frequency  
 (Operation clock to be set by the CKSmn0, CKSmn1 bits of timer mode register mn (TMRmn)  
 m: Unit number (m = 0), n: Channel number (n = 0 to 7))

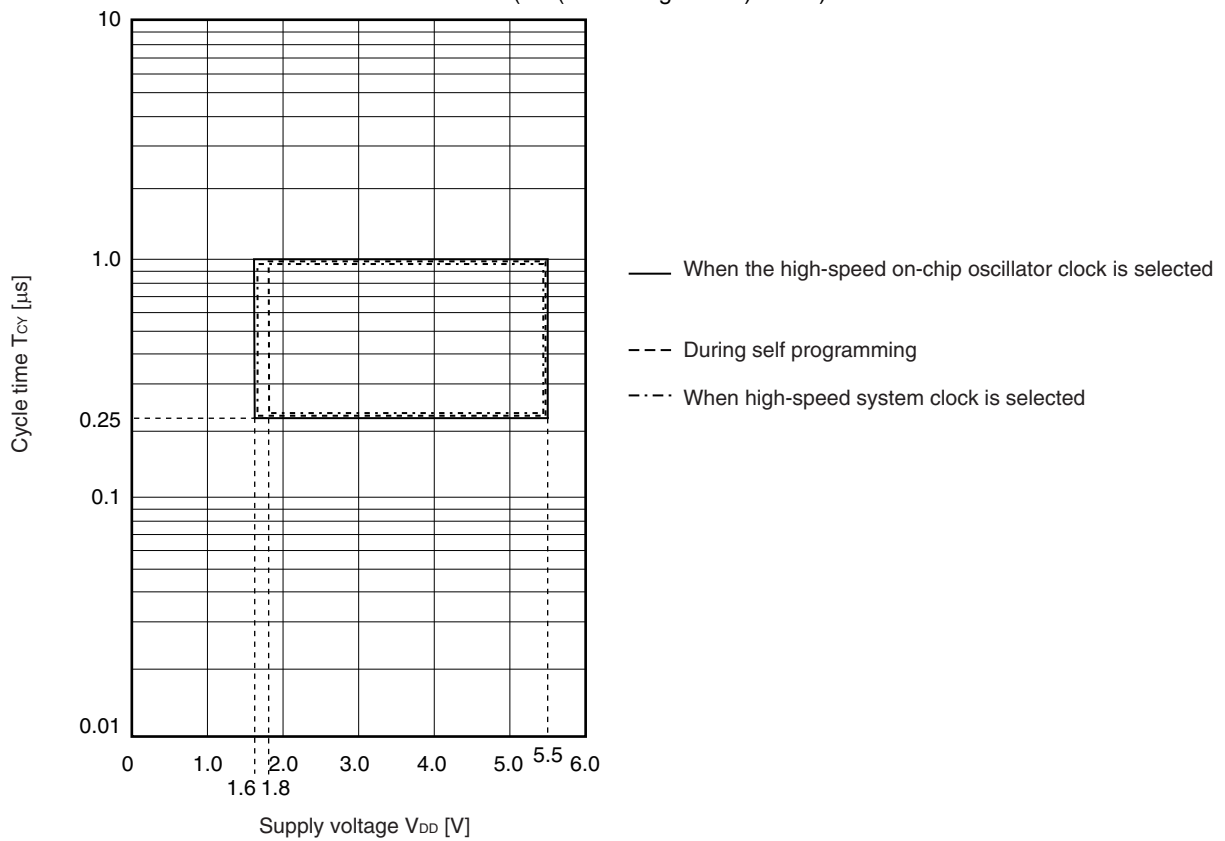
**Minimum Instruction Execution Time during Main System Clock Operation**



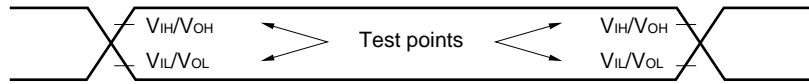
T<sub>CY</sub> vs V<sub>DD</sub> (LS (low-speed main) mode)



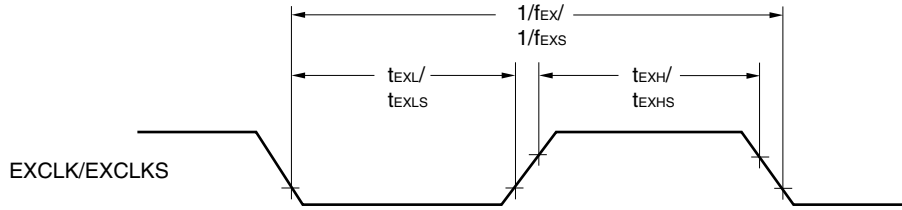
T<sub>CY</sub> vs V<sub>DD</sub> (LV (low-voltage main) mode)



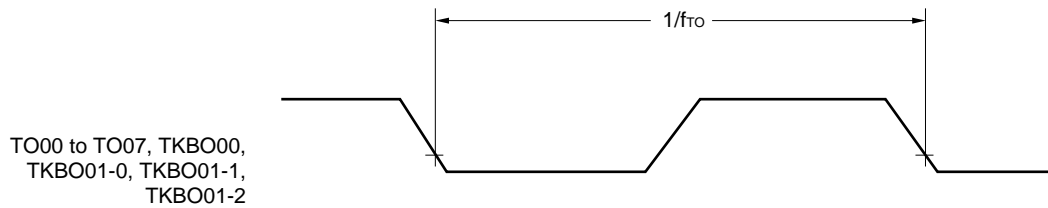
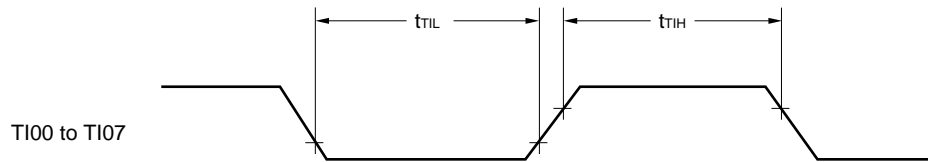
**AC Timing Test Points**



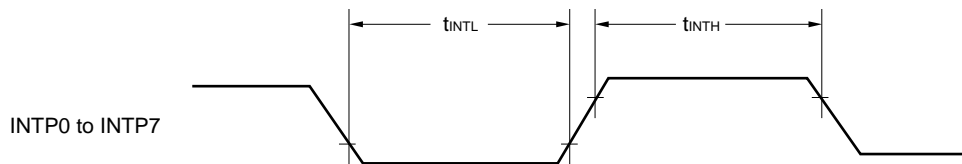
**External System Clock Timing**



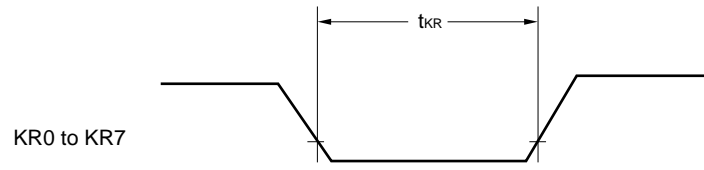
**TI/TO Timing**



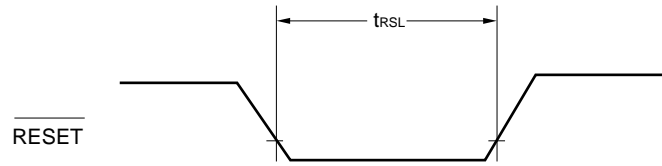
**Interrupt Request Input Timing**



**Key Interrupt Input Timing**

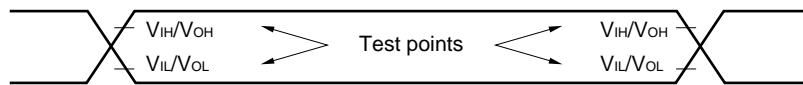


**$\overline{\text{RESET}}$  Input Timing**



### 32.5 Peripheral Functions Characteristics

#### AC Timing Test Points



#### 32.5.1 Serial array unit

##### (1) During communication at same potential (UART mode)

(T<sub>A</sub> = -40 to +85°C, 1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Transfer rate <sup>Note 1</sup>		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		f <sub>MCK</sub> /6		f <sub>MCK</sub> /6		f <sub>MCK</sub> /6	bps
		Theoretical value of the maximum transfer rate f <sub>MCK</sub> = f <sub>CLK</sub> <sup>Note 2</sup>		4.0		1.3		0.6	Mbps
		1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V		–		f <sub>MCK</sub> /6		f <sub>MCK</sub> /6	bps
		Theoretical value of the maximum transfer rate f <sub>MCK</sub> = f <sub>CLK</sub> <sup>Note 2</sup>		–		1.3		0.6	Mbps
		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V		–		–		f <sub>MCK</sub> /6	bps
		Theoretical value of the maximum transfer rate f <sub>MCK</sub> = f <sub>CLK</sub> <sup>Note 2</sup>		–		–		0.6	Mbps

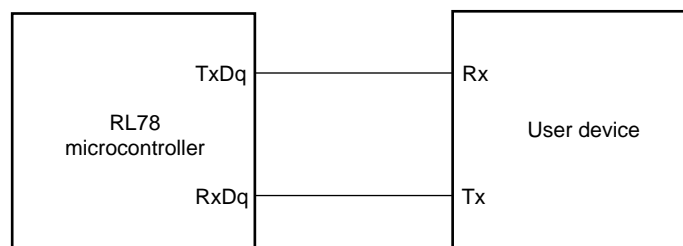
**Notes 1.** Transfer rate in the SNOOZE mode is 4800 bps only.

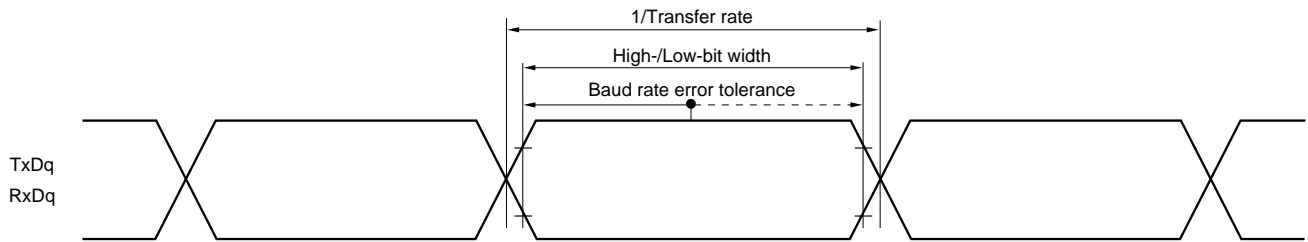
**2.** The maximum operating frequencies of the CPU/peripheral hardware clock (f<sub>CLK</sub>) are:

- HS (high-speed main) mode: 24 MHz (2.7 V ≤ V<sub>DD</sub> ≤ 5.5 V)  
16 MHz (2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V)
- LS (low-speed main) mode: 8 MHz (1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V)
- LV (low-voltage main) mode: 4 MHz (1.6 V ≤ V<sub>DD</sub> ≤ 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 to 3), g: PIM and POM number (g = 0, 1, 3)
  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 to 03, 10 to 13))

**(2) During communication at same potential (CSI mode) (master mode, SCKp... internal clock output)**(T<sub>A</sub> = -40 to +85°C, 1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t <sub>KCY1</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	167 <sup>Note 1</sup>		500 <sup>Note 1</sup>		1000 <sup>Note 1</sup>		ns
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V	250 <sup>Note 1</sup>		500 <sup>Note 1</sup>		1000 <sup>Note 1</sup>		ns
		1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V	–		500 <sup>Note 1</sup>		1000 <sup>Note 1</sup>		ns
		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	–		–		1000 <sup>Note 1</sup>		ns
SCKp high-/low-level width	t <sub>KH1</sub> , t <sub>KL1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V	t <sub>KCY1</sub> /2–12		t <sub>KCY1</sub> /2–50		t <sub>KCY1</sub> /2–50		ns
		2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	t <sub>KCY1</sub> /2–18		t <sub>KCY1</sub> /2–50		t <sub>KCY1</sub> /2–50		ns
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V	t <sub>KCY1</sub> /2–38		t <sub>KCY1</sub> /2–50		t <sub>KCY1</sub> /2–50		ns
		1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V	–		t <sub>KCY1</sub> /2–50		t <sub>KCY1</sub> /2–50		ns
		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	–		–		t <sub>KCY1</sub> /2–100		ns
Slp setup time (to SCKp↑) <sup>Note 2</sup>	t <sub>SIK1</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	44		110		110		ns
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V	75		110		110		ns
		1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V	–		110		110		ns
		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	–		–		220		ns
Slp hold time (from SCKp↑) <sup>Note 3</sup>	t <sub>KSI1</sub>	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V	19		19		19		ns
		1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V	–		19		19		ns
		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	–		–		19		ns
Delay time from SCKp↓ to SOp output <sup>Note 4</sup>	t <sub>KSO1</sub>	C = 30 pF <sup>Note 5</sup>	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		25		25	25	ns
			1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V		–		25	25	ns
			1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V		–		–	25	ns

**Notes 1.** The value must also be equal to or more than 2/f<sub>CLK</sub> for CSI00 and equal to or more than 4/f<sub>CLK</sub> for CSI10.

2. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The Slp setup time becomes “to SCKp↓” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
3. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The Slp hold time becomes “from SCKp↓” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
4. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The delay time to SOp output becomes “from SCKp↑” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
5. C is the load capacitance of the SCKp and SOp output lines.

**Caution** Select the normal input buffer for the Slp 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, 10), m: Unit number (m = 0), n: Channel number (n = 0, 2),  
g: PIM and POM numbers (g = 0, 1)

2. f<sub>MCK</sub>: Serial array unit operation clock frequency  
(Operation clock to be set by the CKS<sub>mn</sub> bit of serial mode register mn (SMR<sub>mn</sub>). m: Unit number, n: Channel number (mn = 00, 02))

**(3) During communication at same potential (CSI mode) (slave mode, SCKp... external clock input)**(T<sub>A</sub> = -40 to +85°C, 1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time <sup>Note 5</sup>	tkCY2	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V	f <sub>MCK</sub> > 20 MHz	8/f <sub>MCK</sub>	–	–	–	–	ns	
			f <sub>MCK</sub> ≤ 20 MHz	6/f <sub>MCK</sub>	–	6/f <sub>MCK</sub>	–	6/f <sub>MCK</sub>	ns	
		2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	f <sub>MCK</sub> > 16 MHz	8/f <sub>MCK</sub>	–	–	–	–	ns	
			f <sub>MCK</sub> ≤ 16 MHz	6/f <sub>MCK</sub>	–	6/f <sub>MCK</sub>	–	6/f <sub>MCK</sub>	ns	
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		6/f <sub>MCK</sub> and 500	–	6/f <sub>MCK</sub>	–	6/f <sub>MCK</sub>	ns	
		1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V		–	–	6/f <sub>MCK</sub>	–	6/f <sub>MCK</sub>	ns	
1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V		–	–	–	–	6/f <sub>MCK</sub>	–	ns		
SCKp high-/low-level width	tkH2, tkL2	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V		tkCY2/2-7	–	tkCY2/2-7	–	tkCY2/2-7	ns	
		2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V		tkCY2/2-8	–	tkCY2/2-8	–	tkCY2/2-8	ns	
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		tkCY2/2-18	–	tkCY2/2-18	–	tkCY2/2-18	ns	
		1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V		–	–	tkCY2/2-18	–	tkCY2/2-18	ns	
		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V		–	–	–	–	tkCY2/2-66	–	ns
Slp setup time (to SCKp↑) <sup>Note 1</sup>	tsIK2	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V		1/f <sub>MCK</sub> +20	–	1/f <sub>MCK</sub> +30	–	1/f <sub>MCK</sub> +30	ns	
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		1/f <sub>MCK</sub> +30	–	1/f <sub>MCK</sub> +30	–	1/f <sub>MCK</sub> +30	ns	
		1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V		–	–	1/f <sub>MCK</sub> +30	–	1/f <sub>MCK</sub> +30	ns	
		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V		–	–	–	–	1/f <sub>MCK</sub> +40	–	ns
Slp hold time (from SCKp↑) <sup>Note 2</sup>	tsIH2	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		1/f <sub>MCK</sub> +31	–	1/f <sub>MCK</sub> +31	–	1/f <sub>MCK</sub> +31	ns	
		1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V		–	–	1/f <sub>MCK</sub> +31	–	1/f <sub>MCK</sub> +31	ns	
		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V		–	–	–	–	1/f <sub>MCK</sub> +250	–	ns
Delay time from SCKp↓ to SOP output <sup>Note 3</sup>	tkSO2	C = 30 pF <sup>Note 4</sup>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V		2/f <sub>MCK</sub> +44	–	2/f <sub>MCK</sub> +110	–	2/f <sub>MCK</sub> +110	ns
			2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		2/f <sub>MCK</sub> +75	–	2/f <sub>MCK</sub> +110	–	2/f <sub>MCK</sub> +110	ns
			1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V		–	–	2/f <sub>MCK</sub> +110	–	2/f <sub>MCK</sub> +110	ns
			1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V		–	–	–	–	2/f <sub>MCK</sub> +220	–

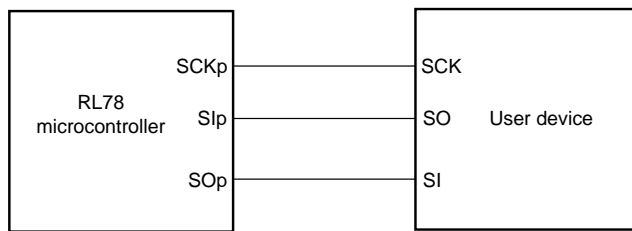
- Notes**
1. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The Slp setup time becomes “to SCKp↓” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
  2. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The Slp hold time becomes “from SCKp↓” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
  3. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The delay time to SOP output becomes “from SCKp↑” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
  4. C is the load capacitance of the SOP output lines.
  5. Transfer rate in SNOOZE mode: MAX. 1 Mbps

**Caution** Select the normal input buffer for the Slp 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).

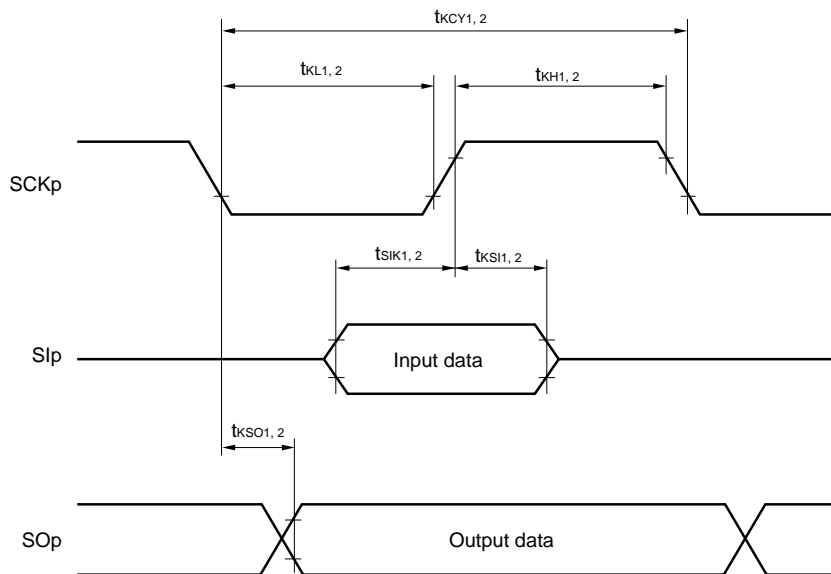
- Remarks**
1. p: CSI number (p = 00, 10), m: Unit number (m = 0), n: Channel number (n = 0, 2), g: PIM number (g = 0, 1)
  2. f<sub>MCK</sub>: Serial array unit operation clock frequency  
(Operation clock to be set by the CKS<sub>mn</sub> bit of serial mode register mn (SMR<sub>mn</sub>). m: Unit number, n: Channel number (mn = 00, 02))



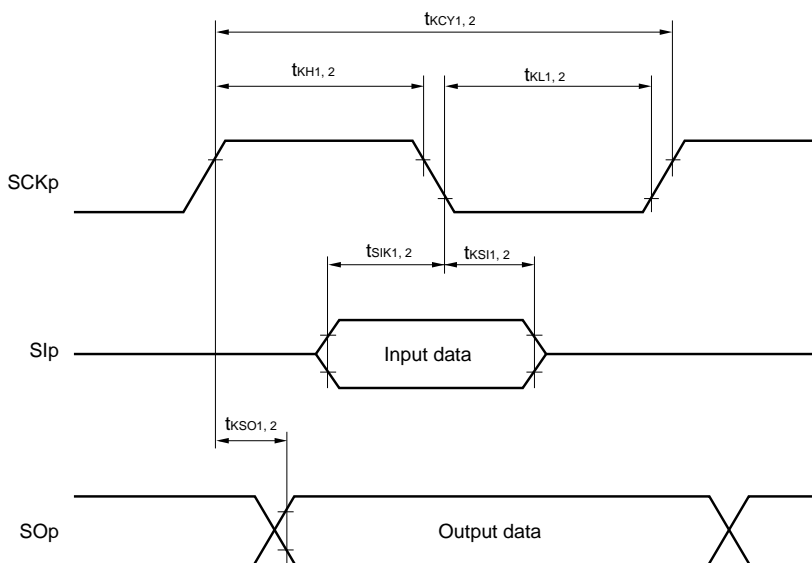
**CSI mode connection diagram (during communication at same potential)**



**CSI mode serial transfer timing (during communication at same potential)  
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)**



**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, 10)
  2. m: Unit number, n: Channel number (mn = 00, 02)

**(4) During communication at same potential (simplified I<sup>2</sup>C mode)****(T<sub>A</sub> = -40 to +85°C, 1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)**

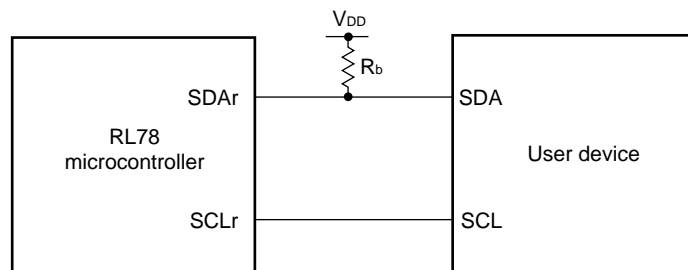
Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLr clock frequency	f <sub>SCL</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ		1000 <sup>Note 1</sup>		400 <sup>Note 1</sup>		400 <sup>Note 1</sup>	kHz
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ		400 <sup>Note 1</sup>		400 <sup>Note 1</sup>		400 <sup>Note 1</sup>	kHz
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> < 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ		300 <sup>Note 1</sup>		300 <sup>Note 1</sup>		300 <sup>Note 1</sup>	kHz
		1.6 V ≤ V <sub>DD</sub> < 1.8 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ		–		–		250 <sup>Note 1</sup>	kHz
Hold time when SCLr = “L”	t <sub>LOW</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	475		1150		1150		ns
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	1150		1150		1150		ns
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> < 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	1550		1550		1550		ns
		1.6 V ≤ V <sub>DD</sub> < 1.8 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	–		–		1850		ns
Hold time when SCLr = “H”	t <sub>HIGH</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	475		1150		1150		ns
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	1150		1150		1150		ns
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> < 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	1550		1550		1550		ns
		1.6 V ≤ V <sub>DD</sub> < 1.8 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	–		–		1850		ns
Data setup time (reception)	t <sub>SU:DAT</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	1/f <sub>MCK+</sub> 85 <sup>Note 2</sup>		1/f <sub>MCK+</sub> 145 <sup>Note 2</sup>		1/f <sub>MCK+</sub> 145 <sup>Note 2</sup>		ns
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	1/f <sub>MCK+</sub> 145 <sup>Note 2</sup>		1/f <sub>MCK+</sub> 145 <sup>Note 2</sup>		1/f <sub>MCK+</sub> 145 <sup>Note 2</sup>		ns
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> < 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	1/f <sub>MCK+</sub> 230 <sup>Note 2</sup>		1/f <sub>MCK+</sub> 230 <sup>Note 2</sup>		1/f <sub>MCK+</sub> 230 <sup>Note 2</sup>		ns
		1.6 V ≤ V <sub>DD</sub> < 1.8 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	–		–		1/f <sub>MCK+</sub> 290 <sup>Note 2</sup>		ns
Data hold time (transmission)	t <sub>HD:DAT</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	0	305	0	305	0	305	ns
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	0	355	0	355	0	355	ns
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> < 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	0	405	0	405	0	405	ns
		1.6 V ≤ V <sub>DD</sub> < 1.8 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	–	–	–	–	0	405	ns

**(Notes, Caution, and Remarks are listed on the next page.)**

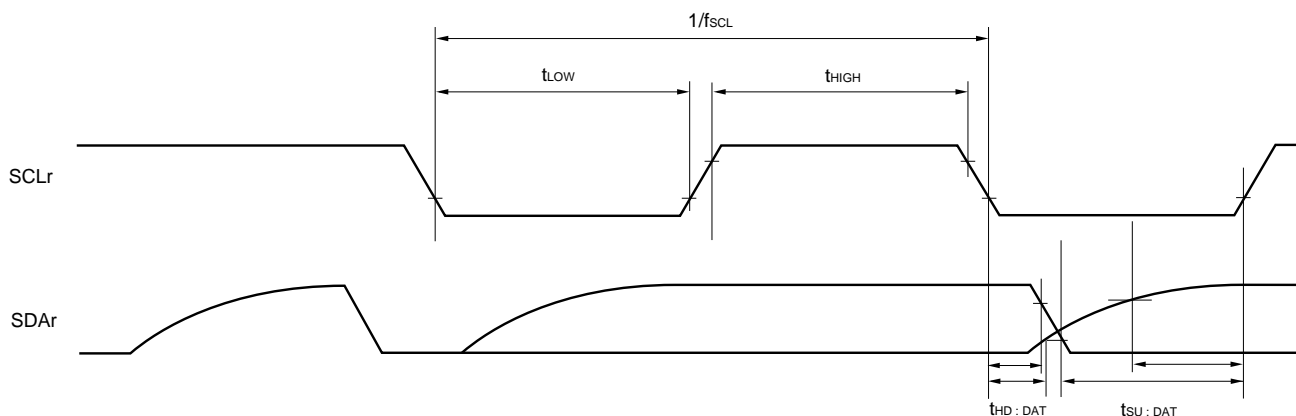
- Notes**
1. The value must also be equal to or less than  $f_{MCK}/4$ .
  2. Set the  $f_{MCK}$  value to keep the hold time of  $SCLr = "L"$  and  $SCLr = "H"$ .
  3. Condition in the HS (high-speed main) mode

**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 g (POMg).

**Simplified I<sup>2</sup>C mode connection diagram (during communication at same potential)**



**Simplified I<sup>2</sup>C mode serial transfer timing (during communication at same potential)**



- Remarks**
1.  $R_b[\Omega]$ : Communication line ( $SDAr$ ) pull-up resistance,  $C_b[F]$ : Communication line ( $SDAr$ ,  $SCLr$ ) load capacitance
  2. r: IIC number (r = 00, 10), g: PIM and POM number (g = 0, 1)
  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 = 0-3), mn = 00-03, 10-13)

<R>

**(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode) (1/2)**(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit	
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.		
Transfer rate		Reception	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V		f <sub>MCK</sub> /6 <sup>Note 1</sup>		f <sub>MCK</sub> /6 <sup>Note 1</sup>		f <sub>MCK</sub> /6 <sup>Note 1</sup>	bps
			Theoretical value of the maximum transfer rate f <sub>MCK</sub> = f <sub>CLK</sub> <sup>Note 3</sup>		4.0		1.3		0.6	Mbps
			2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V		f <sub>MCK</sub> /6 <sup>Note 1</sup>		f <sub>MCK</sub> /6 <sup>Note 1</sup>		f <sub>MCK</sub> /6 <sup>Note 1</sup>	bps
			Theoretical value of the maximum transfer rate f <sub>MCK</sub> = f <sub>CLK</sub> <sup>Note 3</sup>		4.0		1.3		0.6	Mbps
			1.8 V (2.4 V <sup>Note 4</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V		f <sub>MCK</sub> /6 <sup>Note s1, 2</sup>		f <sub>MCK</sub> /6 <sup>Notes 1, 2</sup>		f <sub>MCK</sub> /6 <sup>Notes 1, 2</sup>	bps
			Theoretical value of the maximum transfer rate f <sub>MCK</sub> = f <sub>CLK</sub> <sup>Note 3</sup>		4.0		1.3		0.6	Mbps

**Notes 1.** Transfer rate in SNOOZE mode is 4800 bps only.**2.** Use it with V<sub>DD</sub> ≥ V<sub>b</sub>.**3.** The maximum operating frequencies of the CPU/peripheral hardware clock (f<sub>CLK</sub>) are:HS (high-speed main) mode: 24 MHz (2.7 V ≤ V<sub>DD</sub> ≤ 5.5 V)16 MHz (2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V)LS (low-speed main) mode: 8 MHz (1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V)LV (low-voltage main) mode: 4 MHz (1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V)**4.** Condition in the HS (high-speed main) mode

**Caution** Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V<sub>DD</sub> tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V<sub>IH</sub> and V<sub>IL</sub>, see the DC characteristics with TTL input buffer selected.

**Remarks 1.** V<sub>b</sub>[V]: Communication line voltage**2.** q: UART number (q = 0 to 3), g: PIM and POM number (g = 0, 1, 3)**3.** f<sub>MCK</sub>: 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 to 03, 10 to 13)

**(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode) (2/2)**(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Transfer rate	Transmission	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V		<b>Note 1</b>		<b>Note 1</b>		<b>Note 1</b>	bps
		Theoretical value of the maximum transfer rate (C <sub>b</sub> = 50 pF, R <sub>b</sub> = 1.4 kΩ, V <sub>b</sub> = 2.7 V)		2.8 <sup>Note 2</sup>		2.8 <sup>Note 2</sup>		2.8 <sup>Note 2</sup>	Mbps
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V		<b>Note 3</b>		<b>Note 3</b>		<b>Note 3</b>	bps
		Theoretical value of the maximum transfer rate (C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ, V <sub>b</sub> = 2.3 V)		1.2 <sup>Note 4</sup>		1.2 <sup>Note 4</sup>		1.2 <sup>Note 4</sup>	Mbps
		1.8 V (2.4 V <sup>Note 8</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V		<b>Notes 5, 6</b>		<b>Notes 5, 6</b>		<b>Notes 5, 6</b>	bps
		Theoretical value of the maximum transfer rate (C <sub>b</sub> = 50 pF, R <sub>b</sub> = 5.5 kΩ, V <sub>b</sub> = 1.6 V)		0.43 <sup>Note 7</sup>		0.43 <sup>Note 7</sup>		0.43 <sup>Note 7</sup>	Mbps

**Notes 1.** The smaller maximum transfer rate derived by using f<sub>MCK</sub>/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V ≤ V<sub>DD</sub> ≤ 5.5 V and 2.7 V ≤ V<sub>b</sub> ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

\* This value is the theoretical value of the relative difference between the transmission and reception sides.

- 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.
- The smaller maximum transfer rate derived by using f<sub>MCK</sub>/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ V<sub>DD</sub> < 4.0 V and 2.3 V ≤ V<sub>b</sub> ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

\* This value is the theoretical value of the relative difference between the transmission and reception sides.

- 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.
- Use it with V<sub>DD</sub> ≥ V<sub>b</sub>.

**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  $1.8\text{ V} (2.4\text{ V}^{\text{Note 8}}) \leq V_{DD} < 3.3\text{ V}$  and  $1.6\text{ V} \leq V_b \leq 2.0\text{ V}$

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

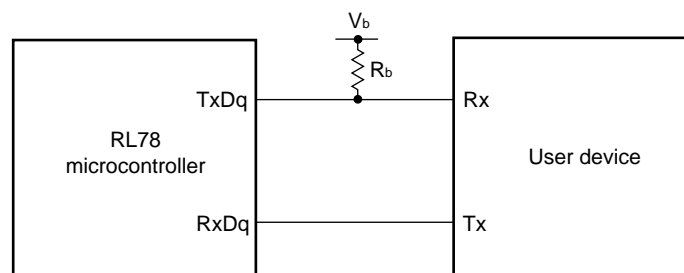
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

\* This value is the theoretical value of the relative difference between the transmission and reception sides.

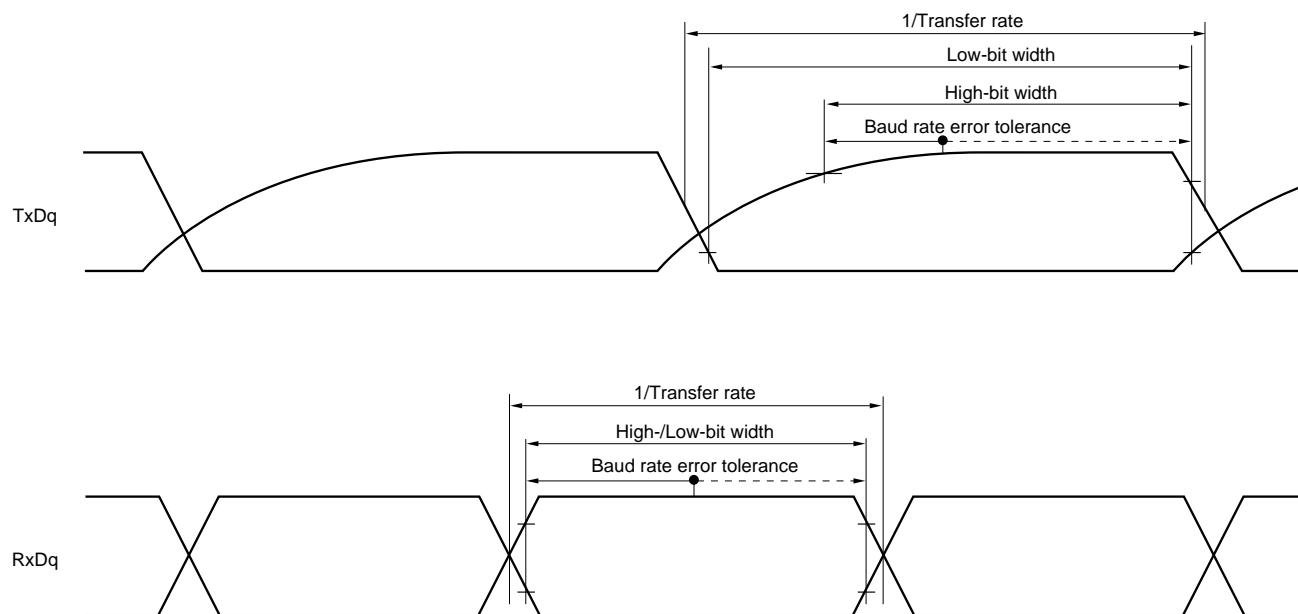
7. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 6** above to calculate the maximum transfer rate under conditions of the customer.
8. Condition in the HS (high-speed main) mode

**Caution** Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V<sub>DD</sub> tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V<sub>IH</sub> and V<sub>IL</sub>, 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.**  $R_b[\Omega]$ : Communication line (TxDq) pull-up resistance,  $C_b[F]$ : Communication line (TxDq) load capacitance,  $V_b[V]$ : Communication line voltage
- 2.**  $q$ : UART number ( $q = 0$  to  $3$ ),  $g$ : PIM and POM number ( $g = 0, 1, 3$ )
- 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 ( $mn = 00$  to  $03, 10$  to  $13$ ))

**(6) Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output, corresponding CSI00 only)****(T<sub>A</sub> = -40 to +85°C, 2.7 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)**

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t <sub>KCY1</sub>	t <sub>KCY1</sub> ≥ 2/f <sub>CLK</sub> 4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 1.4 kΩ	200		1150		1150		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 2.7 kΩ	300		1150		1150		ns
SCKp high-level width	t <sub>KH1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 1.4 kΩ	t <sub>KCY1</sub> /2 – 50		t <sub>KCY1</sub> /2 – 50		t <sub>KCY1</sub> /2 – 50		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 2.7 kΩ	t <sub>KCY1</sub> /2 – 120		t <sub>KCY1</sub> /2 – 120		t <sub>KCY1</sub> /2 – 120		ns
SCKp low-level width	t <sub>KL1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 1.4 kΩ	t <sub>KCY1</sub> /2 – 7		t <sub>KCY1</sub> /2 – 50		t <sub>KCY1</sub> /2 – 50		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 2.7 kΩ	t <sub>KCY1</sub> /2 – 10		t <sub>KCY1</sub> /2 – 50		t <sub>KCY1</sub> /2 – 50		ns
Slp setup time (to SCKp↑) <sup>Note 1</sup>	t <sub>SIK1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 1.4 kΩ	58		479		479		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 2.7 kΩ	121		479		479		ns
Slp hold time (from SCKp↑) <sup>Note 1</sup>	t <sub>KSI1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 1.4 kΩ	10		10		10		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 2.7 kΩ	10		10		10		ns
Delay time from SCKp↓ to SOP output <sup>Note 1</sup>	t <sub>KSO1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 1.4 kΩ		60		60		60	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 2.7 kΩ		130		130		130	ns
Slp setup time (to SCKp↓) <sup>Note 2</sup>	t <sub>SIK1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 1.4 kΩ	23		110		110		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 2.7 kΩ	33		110		110		ns
Slp hold time (from SCKp↓) <sup>Note 2</sup>	t <sub>KSI1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 1.4 kΩ	10		10		10		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 2.7 kΩ	10		10		10		ns
Delay time from SCKp↑ to SOP output <sup>Note 2</sup>	t <sub>KSO1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 1.4 kΩ		10		10		10	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 2.7 kΩ		10		10		10	ns

**(Notes, Caution and Remarks are listed on the next page.)**



- Notes**
1. When  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 0$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 1$ .
  2. When  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 1$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 0$ .

**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.  $R_b[\Omega]$ : Communication line (SCKp, SOp) pull-up resistance,  $C_b[\text{F}]$ : Communication line (SCKp, SOp) load capacitance,  $V_b[\text{V}]$ : 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 = 1)
  3.  $f_{\text{MCK}}$ : 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 specification is valid only when CSI00's peripheral I/O redirect function is not used.

**(7) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output) (1/2)**  
**(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)**

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t <sub>KCY1</sub>	t <sub>KCY1</sub> ≥ 4/f <sub>CLK</sub> 4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	300		1150		1150		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	500		1150		1150		ns
		1.8 V (2.4 V <sup>Note 1</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 1.8 V <sup>Note 2</sup> , C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	1150		1150		1150		ns
SCKp high-level width	t <sub>KH1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	t <sub>KCY1</sub> /2 – 75		t <sub>KCY1</sub> /2 – 75		t <sub>KCY1</sub> /2 – 75		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	t <sub>KCY1</sub> /2 – 170		t <sub>KCY1</sub> /2 – 170		t <sub>KCY1</sub> /2 – 170		ns
		1.8 V (2.4 V <sup>Note 1</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 2</sup> , C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	t <sub>KCY1</sub> /2 – 458		t <sub>KCY1</sub> /2 – 458		t <sub>KCY1</sub> /2 – 458		ns
SCKp low-level width	t <sub>KL1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	t <sub>KCY1</sub> /2 – 12		t <sub>KCY1</sub> /2 – 50		t <sub>KCY1</sub> /2 – 50		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	t <sub>KCY1</sub> /2 – 18		t <sub>KCY1</sub> /2 – 50		t <sub>KCY1</sub> /2 – 50		ns
		1.8 V (2.4 V <sup>Note 1</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 2</sup> , C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	t <sub>KCY1</sub> /2 – 50		t <sub>KCY1</sub> /2 – 50		t <sub>KCY1</sub> /2 – 50		ns
Slp setup time (to SCKp↑) <sup>Note 3</sup>	t <sub>SIK1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	81		479		479		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	177		479		479		ns
		1.8 V (2.4 V <sup>Note 1</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 2</sup> , C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	479		479		479		ns
Slp hold time (from SCKp↑) <sup>Note 3</sup>	t <sub>KSH1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	19		19		19		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	19		19		19		ns
		1.8 V (2.4 V <sup>Note 1</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 2</sup> , C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	19		19		19		ns
Delay time from SCKp↓ to SOP output <sup>Note 3</sup>	t <sub>KSO1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ		100		100		100	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ		195		195		195	ns
		1.8 V (2.4 V <sup>Note 1</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 2</sup> , C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ		483		483		483	ns

(Notes and Caution are listed on the next page, and Remarks are listed on the page after the next page.)

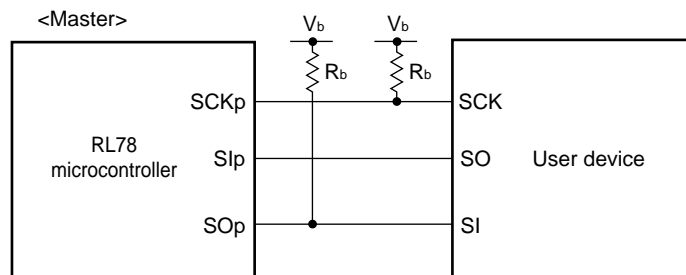
**(7) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output) (2/2)**  
 (T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Slp setup time (to SCKp↓) <sup>Note 4</sup>	t <sub>SIK1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	44		110		110		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	44		110		110		ns
		1.8 V (2.4 V <sup>Note 1</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 2</sup> , C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	110		110		110		ns
Slp hold time (from SCKp↓) <sup>Note 4</sup>	t <sub>KS11</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	19		19		19		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	19		19		19		ns
		1.8 V (2.4 V <sup>Note 1</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 2</sup> , C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	19		19		19		ns
Delay time from SCKp↑ to SOp output <sup>Note 4</sup>	t <sub>KSO1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ		25		25		25	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ		25		25		25	ns
		1.8 V (2.4 V <sup>Note 1</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 2</sup> , C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ		25		25		25	ns

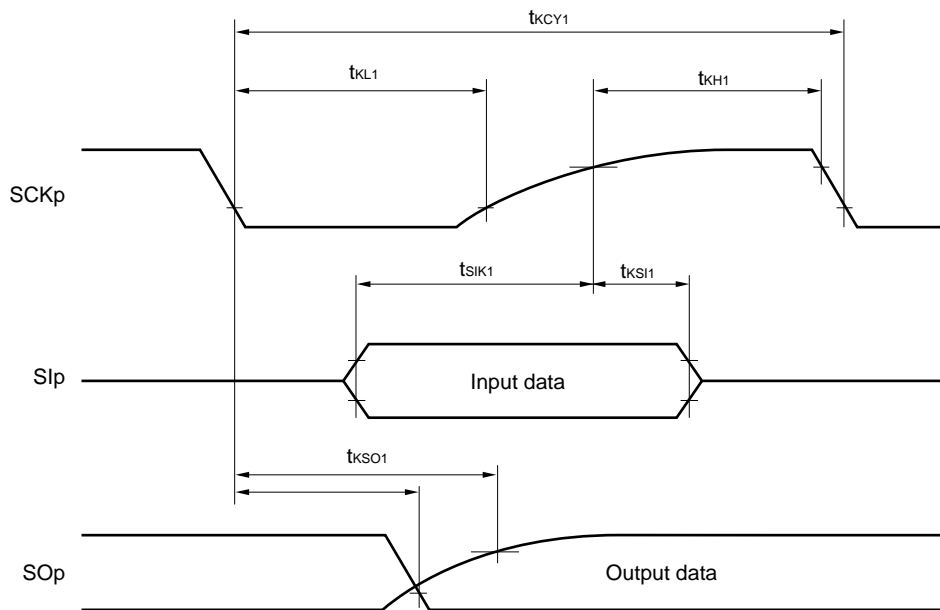
- Notes**
1. Condition in HS (high-speed main) mode
  2. Use it with V<sub>DD</sub> ≥ V<sub>b</sub>.
  3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.
  4. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**Caution** Select the TTL input buffer for the Slp pin and the N-ch open drain output (V<sub>DD</sub> 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<sub>IH</sub> and V<sub>IL</sub>, see the DC characteristics with TTL input buffer selected.

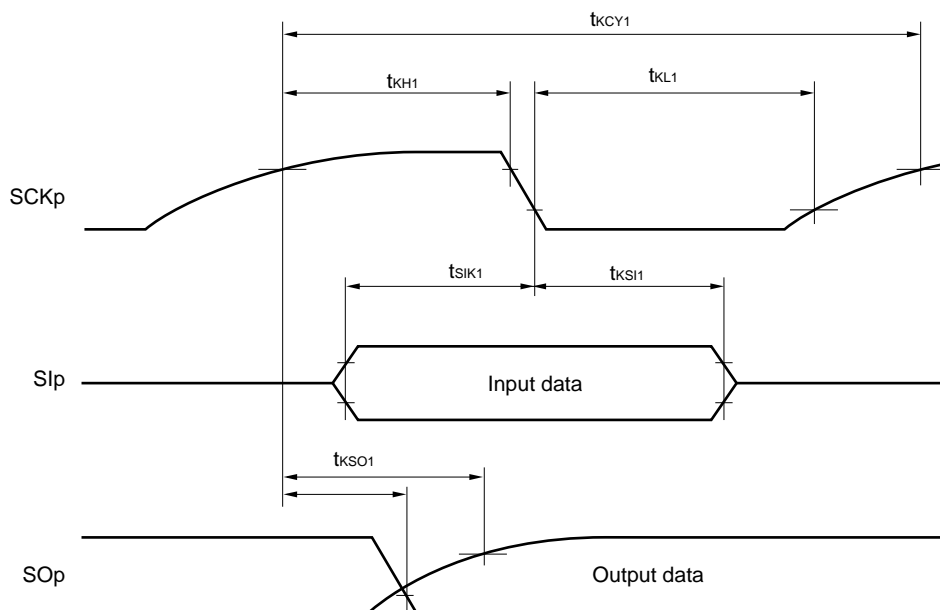
CSI mode connection diagram (during communication at different potential)



**CSI mode serial transfer timing (master mode) (during communication at different potential)  
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)**



**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.**  $R_b[\Omega]$ : 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, 10), m: Unit number, n: Channel number (mn = 00, 02), g: PIM and POM number (g = 0, 1)
- 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)

**(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (slave mode, SCKp... external clock input)**  
**(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)**

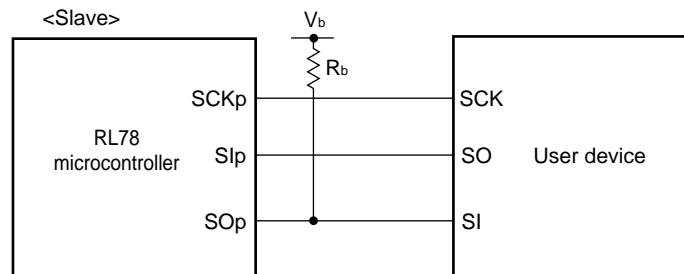
Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time <sup>Note 1</sup>	t <sub>KCY2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V	20 MHz < f <sub>MCK</sub>	12/f <sub>MCK</sub>		–		–		ns
			8 MHz < f <sub>MCK</sub> ≤ 20 MHz	10/f <sub>MCK</sub>		–		–		ns
			4 MHz < f <sub>MCK</sub> ≤ 8 MHz	8/f <sub>MCK</sub>		16/f <sub>MCK</sub>		–		ns
			f <sub>MCK</sub> ≤ 4 MHz	6/f <sub>MCK</sub>		10/f <sub>MCK</sub>		10/f <sub>MCK</sub>		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V	20 MHz < f <sub>MCK</sub>	16/f <sub>MCK</sub>		–		–		ns
			16 MHz < f <sub>MCK</sub> ≤ 20 MHz	14/f <sub>MCK</sub>		–		–		ns
			8 MHz < f <sub>MCK</sub> ≤ 16 MHz	12/f <sub>MCK</sub>		–		–		ns
			4 MHz < f <sub>MCK</sub> ≤ 8 MHz	8/f <sub>MCK</sub>		16/f <sub>MCK</sub>		–		ns
		1.8 V (2.4 V <sup>Note 2</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 3</sup>	20 MHz < f <sub>MCK</sub>	36/f <sub>MCK</sub>		–		–		ns
			16 MHz < f <sub>MCK</sub> ≤ 20 MHz	32/f <sub>MCK</sub>		–		–		ns
			8 MHz < f <sub>MCK</sub> ≤ 16 MHz	26/f <sub>MCK</sub>		–		–		ns
			4 MHz < f <sub>MCK</sub> ≤ 8 MHz	16/f <sub>MCK</sub>		16/f <sub>MCK</sub>		–		ns
f <sub>MCK</sub> ≤ 4 MHz		10/f <sub>MCK</sub>		10/f <sub>MCK</sub>		10/f <sub>MCK</sub>		ns		
	SCKp high-/low-level width		t <sub>KH2</sub> , t <sub>KL2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V		t <sub>KCY2</sub> /2 – 12	t <sub>KCY2</sub> /2 – 50	t <sub>KCY2</sub> /2 – 50		ns
	2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V			t <sub>KCY2</sub> /2 – 18	t <sub>KCY2</sub> /2 – 50	t <sub>KCY2</sub> /2 – 50		ns		
	1.8 V (2.4 V <sup>Note 2</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 3</sup>			t <sub>KCY2</sub> /2 – 50	t <sub>KCY2</sub> /2 – 50	t <sub>KCY2</sub> /2 – 50		ns		
	Slp setup time (to SCKp↑) <sup>Note 4</sup>	t <sub>SIK2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V		1/f <sub>MCK</sub> + 20		1/f <sub>MCK</sub> + 30		1/f <sub>MCK</sub> + 30	ns
2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V			1/f <sub>MCK</sub> + 20		1/f <sub>MCK</sub> + 30		1/f <sub>MCK</sub> + 30	ns		
1.8 V (2.4 V <sup>Note 2</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 3</sup>			1/f <sub>MCK</sub> + 30		1/f <sub>MCK</sub> + 30		1/f <sub>MCK</sub> + 30	ns		
Slp hold time (from SCKp↑) <sup>Note 5</sup>	t <sub>SI2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V		1/f <sub>MCK</sub> + 31		1/f <sub>MCK</sub> + 31		1/f <sub>MCK</sub> + 31	ns	
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V		1/f <sub>MCK</sub> + 31		1/f <sub>MCK</sub> + 31		1/f <sub>MCK</sub> + 31	ns	
		1.8 V (2.4 V <sup>Note 2</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 3</sup>		1/f <sub>MCK</sub> + 31		1/f <sub>MCK</sub> + 31		1/f <sub>MCK</sub> + 31	ns	
Delay time from SCKp↓ to SOP output <sup>Note 6</sup>	t <sub>KS02</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ			2/f <sub>MCK</sub> + 120		2/f <sub>MCK</sub> + 573		2/f <sub>MCK</sub> + 573	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ			2/f <sub>MCK</sub> + 214		2/f <sub>MCK</sub> + 573		2/f <sub>MCK</sub> + 573	ns
		1.8 V (2.4 V <sup>Note 2</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 3</sup> , C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ			2/f <sub>MCK</sub> + 573		2/f <sub>MCK</sub> + 573		2/f <sub>MCK</sub> + 573	ns

(Notes and Caution are listed on the next page, and Remarks are listed on the page after the next page.)

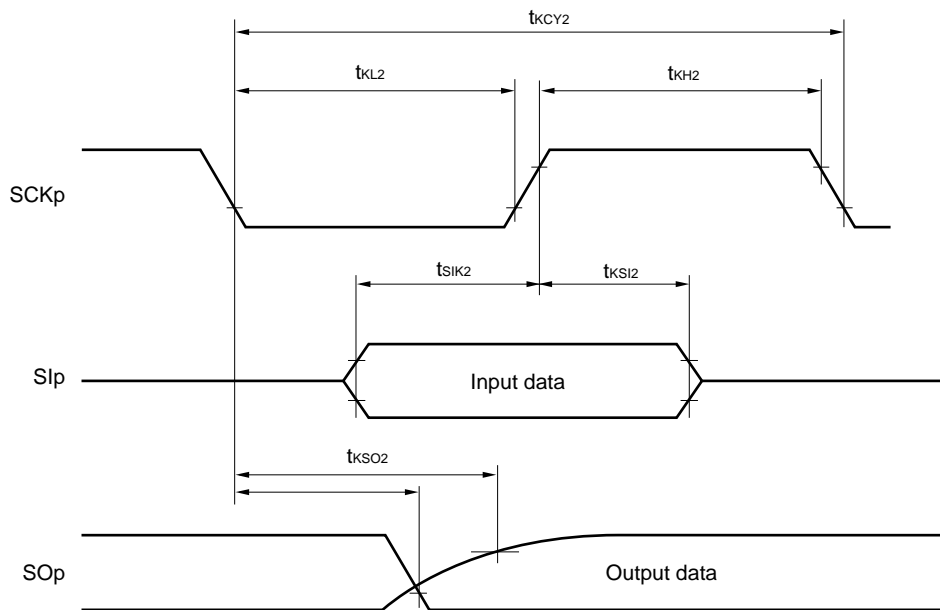
- Notes**
1. Transfer rate in SNOOZE mode: MAX. 1 Mbps
  2. Condition in HS (high-speed main) mode
  3. Use it with  $V_{DD} \geq V_b$ .
  4. When  $DAPmn = 0$  and  $CKPmn = 0$ , or  $DAPmn = 1$  and  $CKPmn = 1$ . The  $Slp$  setup time becomes “to  $SCKp\downarrow$ ” when  $DAPmn = 0$  and  $CKPmn = 1$ , or  $DAPmn = 1$  and  $CKPmn = 0$ .
  5. When  $DAPmn = 0$  and  $CKPmn = 0$ , or  $DAPmn = 1$  and  $CKPmn = 1$ . The  $Slp$  hold time becomes “from  $SCKp\downarrow$ ” when  $DAPmn = 0$  and  $CKPmn = 1$ , or  $DAPmn = 1$  and  $CKPmn = 0$ .
  6. When  $DAPmn = 0$  and  $CKPmn = 0$ , or  $DAPmn = 1$  and  $CKPmn = 1$ . The delay time to  $SOp$  output becomes “from  $SCKp\uparrow$ ” when  $DAPmn = 0$  and  $CKPmn = 1$ , or  $DAPmn = 1$  and  $CKPmn = 0$ .

**Caution** Select the TTL input buffer for the  $Slp$  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.

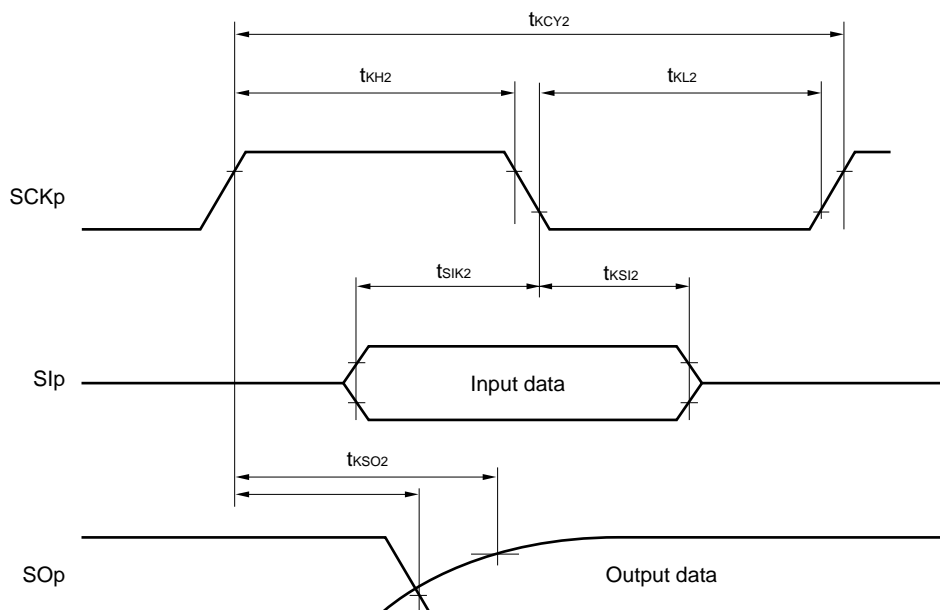
**CSI mode connection diagram (during communication at different potential)**



**CSI mode serial transfer timing (slave mode) (during communication at different potential)  
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)**



**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.** R<sub>b</sub>[Ω]: Communication line (SO<sub>p</sub>) pull-up resistance, C<sub>b</sub>[F]: Communication line (SO<sub>p</sub>) load capacitance, V<sub>b</sub>[V]: Communication line voltage
- 2.** p: CSI number (p = 00, 10), m: Unit number, n: Channel number (mn = 00, 02), g: PIM and POM number (g = 0, 1)
- 3.** f<sub>mck</sub>: Serial array unit operation clock frequency  
(Operation clock to be set by the CKS<sub>mn</sub> bit of serial mode register mn (SMR<sub>mn</sub>))  
m: Unit number, n: Channel number (mn = 00, 02))

(9) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I<sup>2</sup>C mode) (1/2) $(T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLr clock frequency	$f_{SCL}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$ , $C_b = 50\text{ pF}$ , $R_b = 2.7\text{ k}\Omega$		1000 <sup>Note 1</sup>		300 <sup>Note 1</sup>		300 <sup>Note 1</sup>	kHz
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$ , $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$ , $C_b = 50\text{ pF}$ , $R_b = 2.7\text{ k}\Omega$		1000 <sup>Note 1</sup>		300 <sup>Note 1</sup>		300 <sup>Note 1</sup>	kHz
		$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$ , $C_b = 100\text{ pF}$ , $R_b = 2.8\text{ k}\Omega$		400 <sup>Note 1</sup>		300 <sup>Note 1</sup>		300 <sup>Note 1</sup>	kHz
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$ , $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$ , $C_b = 100\text{ pF}$ , $R_b = 2.7\text{ k}\Omega$		400 <sup>Note 1</sup>		300 <sup>Note 1</sup>		300 <sup>Note 1</sup>	kHz
		$1.8\text{ V}$ ( $2.4\text{ V}^{\text{Note 2}}$ ) $\leq V_{DD} < 3.3\text{ V}$ , $1.6\text{ V} \leq V_b \leq 2.0\text{ V}^{\text{Note 3}}$ , $C_b = 100\text{ pF}$ , $R_b = 5.5\text{ k}\Omega$		300 <sup>Note 1</sup>		300 <sup>Note 1</sup>		300 <sup>Note 1</sup>	kHz
Hold time when SCLr = "L"	$t_{LOW}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$ , $C_b = 50\text{ pF}$ , $R_b = 2.7\text{ k}\Omega$	475		1550		1550		ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$ , $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$ , $C_b = 50\text{ pF}$ , $R_b = 2.7\text{ k}\Omega$	475		1550		1550		ns
		$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$ , $C_b = 100\text{ pF}$ , $R_b = 2.8\text{ k}\Omega$	1150		1550		1550		ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$ , $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$ , $C_b = 100\text{ pF}$ , $R_b = 2.7\text{ k}\Omega$	1150		1550		1550		ns
		$1.8\text{ V}$ ( $2.4\text{ V}^{\text{Note 2}}$ ) $\leq V_{DD} < 3.3\text{ V}$ , $1.6\text{ V} \leq V_b \leq 2.0\text{ V}^{\text{Note 3}}$ , $C_b = 100\text{ pF}$ , $R_b = 5.5\text{ k}\Omega$	1550		1550		1550		ns
Hold time when SCLr = "H"	$t_{HIGH}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$ , $C_b = 50\text{ pF}$ , $R_b = 2.7\text{ k}\Omega$	245		610		610		ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$ , $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$ , $C_b = 50\text{ pF}$ , $R_b = 2.7\text{ k}\Omega$	200		610		610		ns
		$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$ , $C_b = 100\text{ pF}$ , $R_b = 2.8\text{ k}\Omega$	675		610		610		ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$ , $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$ , $C_b = 100\text{ pF}$ , $R_b = 2.7\text{ k}\Omega$	600		610		610		ns
		$1.8\text{ V}$ ( $2.4\text{ V}^{\text{Note 2}}$ ) $\leq V_{DD} < 3.3\text{ V}$ , $1.6\text{ V} \leq V_b \leq 2.0\text{ V}^{\text{Note 3}}$ , $C_b = 100\text{ pF}$ , $R_b = 5.5\text{ k}\Omega$	610		610		610		ns

(Notes and Caution are listed on the next page, and Remarks are listed on the page after the next page.)



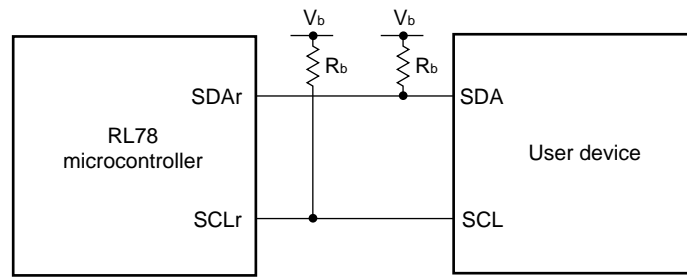
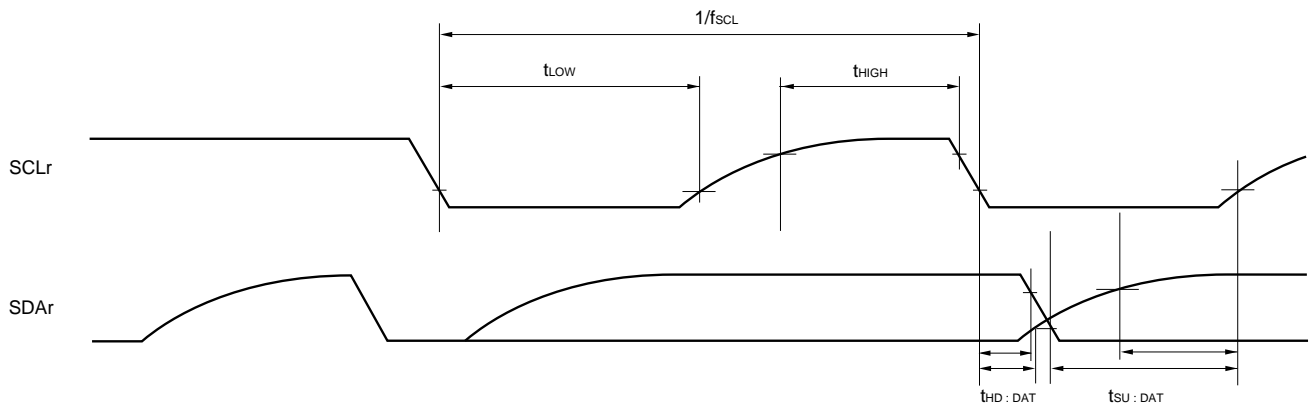
**(9) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I<sup>2</sup>C mode) (2/2)****(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)**

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Data setup time (reception)	t <sub>SU:DAT</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	1/f <sub>MCK</sub> + 135 <sup>Note 4</sup>		1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	1/f <sub>MCK</sub> + 135 <sup>Note 4</sup>		1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		ns
		4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.8 kΩ	1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		ns
		1.8 V (2.4 V <sup>Note 2</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 3</sup> , C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5.5 kΩ	1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		1/f <sub>MCK</sub> + 190 <sup>Note 4</sup>		ns
Data hold time (transmission)	t <sub>HD:DAT</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	0	305	0	305	0	305	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	0	305	0	305	0	305	ns
		4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.8 kΩ	0	355	0	355	0	355	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	0	355	0	355	0	355	ns
		1.8 V (2.4 V <sup>Note 2</sup> ) ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V <sup>Note 3</sup> , C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5.5 kΩ	0	405	0	405	0	405	ns

- Notes**
1. The value must also be equal to or less than f<sub>MCK</sub>/4.
  2. Condition in HS (high-speed main) mode
  3. Use it with V<sub>DD</sub> ≥ V<sub>b</sub>.
  4. Set the f<sub>MCK</sub> 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<sub>DD</sub> tolerance) mode for the SDAr pin and the N-ch open drain output (V<sub>DD</sub> tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V<sub>IH</sub> and V<sub>IL</sub>, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

**Simplified I<sup>2</sup>C mode connection diagram (during communication at different potential)****Simplified I<sup>2</sup>C mode serial transfer timing (during communication at different potential)**

- Remarks 1.**  $R_b[\Omega]$ : 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, 10), g: PIM, POM number (g = 0, 1)
- 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, 02)

## 32.5.2 Serial interface IICA

(1) I<sup>2</sup>C standard mode (1/2)(T<sub>A</sub> = -40 to +85°C, 1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit	
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.		
SCLA0 clock frequency	f <sub>SCL</sub>	Normal mode: f <sub>CLK</sub> ≥ 1 MHz	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0	100	0	100	0	100	kHz
			1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	0	100	0	100	0	100	kHz
			1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	–	–	–	–	0	100	kHz
Setup time of restart condition	t <sub>SU:STA</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	4.7		4.7		4.7		μs	
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	4.7		4.7		4.7		μs	
		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	–	–	–	–	4.7		μs	
Hold time <sup>Note 1</sup>	t <sub>HD:STA</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	4.0		4.0		4.0		μs	
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	4.0		4.0		4.0		μs	
		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	–	–	–	–	4.0		μs	
Hold time when SCLA0 = "L"	t <sub>LOW</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	4.7		4.7		4.7		μs	
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	4.7		4.7		4.7		μs	
		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	–	–	–	–	4.7		μs	
Hold time when SCLA0 = "H"	t <sub>HIGH</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	4.0		4.0		4.0		μs	
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	4.0		4.0		4.0		μs	
		1.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	–	–	–	–	4.0		μs	

(Notes, Caution and Remark are listed on the next page.)

(1) I<sup>2</sup>C standard mode (2/2) $(T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Data setup time (reception)	t <sub>SU:DAT</sub>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	250		250		250		ns
		$1.8\text{ V} (2.4\text{ V}^{\text{Note 3}}) \leq V_{DD} \leq 5.5\text{ V}$	250		250		250		ns
		$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	–	–	–	–	250		ns
Data hold time (transmission) <sup>Note 2</sup>	t <sub>HD:DAT</sub>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0	3.45	0	3.45	0	3.45	μs
		$1.8\text{ V} (2.4\text{ V}^{\text{Note 3}}) \leq V_{DD} \leq 5.5\text{ V}$	0	3.45	0	3.45	0	3.45	μs
		$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	–	–	–	–	0	3.45	μs
Setup time of stop condition	t <sub>SU:STO</sub>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	4.0		4.0		4.0		μs
		$1.8\text{ V} (2.4\text{ V}^{\text{Note 3}}) \leq V_{DD} \leq 5.5\text{ V}$	4.0		4.0		4.0		μs
		$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	–	–	–	–	4.0		μs
Bus-free time	t <sub>BUF</sub>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	4.7		4.7		4.7		μs
		$1.8\text{ V} (2.4\text{ V}^{\text{Note 3}}) \leq V_{DD} \leq 5.5\text{ V}$	4.7		4.7		4.7		μs
		$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	–	–	–	–	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<sub>HD:DAT</sub> is during normal transfer and a wait state is inserted in the ACK (acknowledge) timing.
  3. Condition in HS (high-speed main) mode

**Caution** The values in the above table are applied even when bit 2 (PIOR2) in the peripheral I/O redirection register (PIOR) is 1. At this time, the pin characteristics (I<sub>OH1</sub>, I<sub>OL1</sub>, V<sub>OH1</sub>, V<sub>OL1</sub>) must satisfy the values in the redirect destination.

**Remark** The maximum value of C<sub>b</sub> (communication line capacitance) and the value of R<sub>b</sub> (communication line pull-up resistor) at that time in each mode are as follows.

Standard mode: C<sub>b</sub> = 400 pF, R<sub>b</sub> = 2.7 kΩ

(2) I<sup>2</sup>C fast mode(T<sub>A</sub> = -40 to +85°C, 1.6 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit	
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.		
SCLA0 clock frequency	f <sub>SCL</sub>	Fast mode: f <sub>CLK</sub> ≥ 3.5 MHz	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0	400	0	400	0	400	kHz
			1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	0	400	0	400	0	400	kHz
Setup time of restart condition	t <sub>SU:STA</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.6		0.6		0.6		μs	
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	0.6		0.6		0.6		μs	
Hold time <sup>Note 1</sup>	t <sub>HD:STA</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.6		0.6		0.6		μs	
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	0.6		0.6		0.6		μs	
Hold time when SCLA0 = "L"	t <sub>LOW</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	1.3		1.3		1.3		μs	
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	1.3		1.3		1.3		μs	
Hold time when SCLA0 = "H"	t <sub>HIGH</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.6		0.6		0.6		μs	
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	0.6		0.6		0.6		μs	
Data setup time (reception)	t <sub>SU:DAT</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	100		100		100		ns	
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	100		100		100		ns	
Data hold time (transmission) <sup>Note 2</sup>	t <sub>HD:DAT</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0	0.9	0	0.9	0	0.9	μs	
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	0	0.9	0	0.9	0	0.9	μs	
Setup time of stop condition	t <sub>SU:STO</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.6		0.6		0.6		μs	
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	0.6		0.6		0.6		μs	
Bus-free time	t <sub>BUF</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	1.3		1.3		1.3		μs	
		1.8 V (2.4 V <sup>Note 3</sup> ) ≤ V <sub>DD</sub> ≤ 5.5 V	1.3		1.3		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<sub>HD:DAT</sub> is during normal transfer and a wait state is inserted in the ACK (acknowledge) timing.
  3. Condition in HS (high-speed main) mode

**Caution** The values in the above table are applied even when bit 2 (PIOR2) in the peripheral I/O redirection register (PIOR) is 1. At this time, the pin characteristics (I<sub>OH1</sub>, I<sub>OL1</sub>, V<sub>OH1</sub>, V<sub>OL1</sub>) must satisfy the values in the redirect destination.

**Remark** The maximum value of C<sub>b</sub> (communication line capacitance) and the value of R<sub>b</sub> (communication line pull-up resistor) at that time in each mode are as follows.

Fast mode: C<sub>b</sub> = 320 pF, R<sub>b</sub> = 1.1 kΩ

**(3) I<sup>2</sup>C fast mode plus****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f <sub>SCL</sub>	Fast mode plus: f <sub>CLK</sub> ≥ 10 MHz	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0	1000	–	–	–	–	kHz
Setup time of restart condition	t <sub>SU:STA</sub>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.26		–	–	–	–	μs
Hold time <sup>Note 1</sup>	t <sub>HD:STA</sub>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.26		–	–	–	–	μs
Hold time when SCLA0 = "L"	t <sub>LOW</sub>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.5		–	–	–	–	μs
Hold time when SCLA0 = "H"	t <sub>HIGH</sub>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.26		–	–	–	–	μs
Data setup time (reception)	t <sub>SU:DAT</sub>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		50		–	–	–	–	ns
Data hold time (transmission) <sup>Note 2</sup>	t <sub>HD:DAT</sub>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0	0.45	–	–	–	–	μs
Setup time of stop condition	t <sub>SU:STO</sub>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.26		–	–	–	–	μs
Bus-free time	t <sub>BUF</sub>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		0.5		–	–	–	–	μs

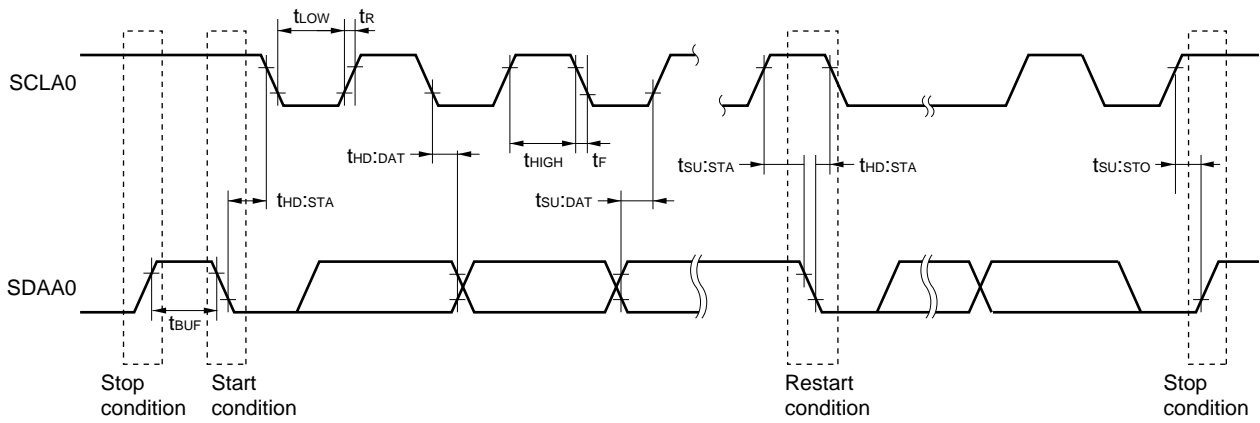
- Notes**
1. The first clock pulse is generated after this period when the start/restart condition is detected.
  2. The maximum value (MAX.) of t<sub>HD:DAT</sub> is during normal transfer and a wait state is inserted in the ACK (acknowledge) timing.

**Caution** The values in the above table are applied even when bit 2 (PIOR2) in the peripheral I/O redirection register (PIOR) is 1. At this time, the pin characteristics (I<sub>OH1</sub>, I<sub>OL1</sub>, V<sub>OH1</sub>, V<sub>OL1</sub>) must satisfy the values in the redirect destination.

**Remark** The maximum value of C<sub>b</sub> (communication line capacitance) and the value of R<sub>b</sub> (communication line pull-up resistor) at that time in each mode are as follows.

Fast mode plus: C<sub>b</sub> = 120 pF, R<sub>b</sub> = 1.1 kΩ

IICA serial transfer timing



## 32.6 Analog Characteristics

### 32.6.1 A/D converter characteristics

#### Classification of A/D converter characteristics

Reference Voltage Input channel	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, ANI1	–	See 32.6.1 (2).	See 32.6.1 (3).
ANI16 to ANI25	See 32.6.1 (1).		
Internal reference voltage Temperature sensor output voltage	See 32.6.1 (1).		–

(1) When reference voltage (+) =  $AV_{REFP}/ANI0$  ( $ADREFP1 = 0$ ,  $ADREFP0 = 1$ ), reference voltage (-) =  $AV_{REFM}/ANI1$  ( $ADREFM = 1$ ), target pins: ANI16 to ANI25, internal reference voltage, and temperature sensor output voltage

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ , Reference voltage (+) =  $AV_{REFP}$ , Reference voltage (-) =  $AV_{REFM} = 0\text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Resolution	RES		8		10	bit	
Overall error <sup>Note 1</sup>	AINL	10-bit resolution $AV_{REFP} = V_{DD}$ <sup>Note 3</sup>	$1.8\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$		1.2	$\pm 5.0$	LSB
			$1.6\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$ <sup>Note 4</sup>		1.2	$\pm 8.5$	LSB
Conversion time	$t_{CONV}$	10-bit resolution Target pin: ANI16 to ANI25	$3.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.125		39	$\mu\text{s}$
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	3.1875		39	$\mu\text{s}$
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	17		39	$\mu\text{s}$
			$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	57		95	$\mu\text{s}$
		10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	$3.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.375		39	$\mu\text{s}$
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	3.5625		39	$\mu\text{s}$
Zero-scale error <sup>Notes 1, 2</sup>	E <sub>ZS</sub>	10-bit resolution $AV_{REFP} = V_{DD}$ <sup>Note 3</sup>	$1.8\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$			$\pm 0.35$	%FSR
			$1.6\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$ <sup>Note 4</sup>			$\pm 0.60$	%FSR
Full-scale error <sup>Notes 1, 2</sup>	E <sub>FS</sub>	10-bit resolution $AV_{REFP} = V_{DD}$ <sup>Note 3</sup>	$1.8\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$			$\pm 0.35$	%FSR
			$1.6\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$ <sup>Note 4</sup>			$\pm 0.60$	%FSR
Integral linearity error <sup>Note 1</sup>	ILE	10-bit resolution $AV_{REFP} = V_{DD}$ <sup>Note 3</sup>	$1.8\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$			$\pm 3.5$	LSB
			$1.6\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$ <sup>Note 4</sup>			$\pm 6.0$	LSB
Differential linearity error <sup>Note 1</sup>	DLE	10-bit resolution $AV_{REFP} = V_{DD}$ <sup>Note 3</sup>	$1.8\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$			$\pm 2.0$	LSB
			$1.6\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$ <sup>Note 4</sup>			$\pm 2.5$	LSB
Analog input voltage	$V_{AIN}$	ANI16 to ANI25	0		$AV_{REFP}$	V	
		Internal reference voltage ( $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , HS (high-speed main) mode))			$V_{BGR}$ <sup>Note 5</sup>	V	
		Temperature sensor output voltage ( $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , HS (high-speed main) mode))			$V_{TMPS25}$ <sup>Note 5</sup>	V	

(Notes are listed on the next page.)



- Notes**
1. Excludes quantization error ( $\pm 1/2$  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 $\pm 4$ LSB to the MAX. value when $AV_{REFP} = V_{DD}$ .
Zero-scale error/Full-scale error:	Add $\pm 0.2\%$ FSR to the MAX. value when $AV_{REFP} = V_{DD}$ .
Integral linearity error/ Differential linearity error:	Add $\pm 2$ LSB to the MAX. value when $AV_{REFP} = V_{DD}$ .
  4. Values when the conversion time is set to  $57 \mu\text{s}$  (min.) and  $95 \mu\text{s}$  (max.).
  5. See **32.6.2 Temperature sensor/internal reference voltage characteristics**.

(2) When reference voltage (+) =  $V_{DD}$  (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) =  $V_{SS}$  (ADREFM = 0), target pins: ANI0, ANI1, ANI16 to ANI25, internal reference voltage, and temperature sensor output voltage

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ , Reference voltage (+) =  $V_{DD}$ , Reference voltage (-) =  $V_{SS}$ )

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error <sup>Notes 1, 2</sup>	AINL	10-bit resolution	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		1.2	$\pm 7.0$	LSB
			$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ <sup>Note 3</sup>		1.2	$\pm 10.5$	LSB
Conversion time	$t_{CONV}$	10-bit resolution Target pin: ANI0, ANI1, ANI16 to ANI25 <sup>Note 3</sup>	$3.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.125		39	$\mu\text{s}$
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	3.1875		39	$\mu\text{s}$
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	17		39	$\mu\text{s}$
			$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	57		95	$\mu\text{s}$
		10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	$3.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.375		39	$\mu\text{s}$
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	3.5625		39	$\mu\text{s}$
$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	17			39	$\mu\text{s}$		
Zero-scale error <sup>Notes 1, 2</sup>	$E_{ZS}$	10-bit resolution	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			$\pm 0.60$	%FSR
			$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ <sup>Note 3</sup>			$\pm 0.85$	%FSR
Full-scale error <sup>Notes 1, 2</sup>	$E_{FS}$	10-bit resolution	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			$\pm 0.60$	%FSR
			$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ <sup>Note 3</sup>			$\pm 0.85$	%FSR
Integral linearity error <sup>Note 1</sup>	ILE	10-bit resolution	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			$\pm 4.0$	LSB
			$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ <sup>Note 3</sup>			$\pm 6.5$	LSB
Differential linearity error <sup>Note 1</sup>	DLE	10-bit resolution	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			$\pm 2.0$	LSB
			$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ <sup>Note 3</sup>			$\pm 2.5$	LSB
Analog input voltage	$V_{AIN}$	ANI0, ANI1, ANI16 to ANI25		0		$V_{DD}$	V
		Internal reference voltage ( $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , HS (high-speed main) mode))		$V_{BGR}$ <sup>Note 4</sup>			V
		Temperature sensor output voltage ( $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , HS (high-speed main) mode))		$V_{TSPS25}$ <sup>Note 4</sup>			V

**Notes** 1. Excludes quantization error ( $\pm 1/2$  LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. Values when the conversion time is set to  $57\ \mu\text{s}$  (min.) and  $95\ \mu\text{s}$  (max.).

4. See 32.6.2 Temperature sensor/internal reference voltage characteristics.

(3) When reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), reference voltage (-) = AVREFM/ANI1 (ADREFM = 1), target pins: ANI0, ANI16 to ANI25

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ , Reference voltage (+) =  $V_{BGR}$ <sup>Note 3</sup>,  
Reference voltage (-) =  $AV_{REFM}$ <sup>Note 4</sup> = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8			bit
Conversion time	t <sub>CONV</sub>	8-bit resolution	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	17		39	$\mu\text{s}$
Zero-scale error <sup>Notes 1, 2</sup>	E <sub>ZS</sub>	8-bit resolution	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			$\pm 0.60$	%FSR
Integral linearity error <sup>Note 1</sup>	ILE	8-bit resolution	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			$\pm 2.0$	LSB
Differential linearity error <sup>Note 1</sup>	DLE	8-bit resolution	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			$\pm 1.0$	LSB
Analog input voltage	V <sub>AIN</sub>			0		$V_{BGR}$ <sup>Note 3</sup>	V

**Notes** 1. Excludes quantization error ( $\pm 1/2$  LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. See 32.6.2 Temperature sensor/internal reference voltage characteristics.

4. When reference voltage (-) =  $V_{SS}$ , the MAX. values are as follows.

Zero-scale error: Add  $\pm 0.35\%$ FSR to the  $AV_{REFM}$  MAX. value.

Integral linearity error: Add  $\pm 0.5$  LSB to the  $AV_{REFM}$  MAX. value.

Differential linearity error: Add  $\pm 0.2$  LSB to the  $AV_{REFM}$  MAX. value.

**32.6.2 Temperature sensor /internal reference voltage characteristics****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ , HS (high-speed main) mode)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	$V_{TMPS25}$	ADS register = 80H, $T_A = +25^\circ\text{C}$		1.05		V
Internal reference output voltage	$V_{BGR}$	ADS register = 81H	1.38	1.45	1.5	V
Temperature coefficient	$F_{VTMPS}$	Temperature sensor that depends on the temperature		-3.6		mV/ $^\circ\text{C}$
Operation stabilization wait time	$t_{AMP}$				5	$\mu\text{s}$

**32.6.3 Comparator characteristics****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input voltage range	$I_{VREF}$		0		$V_{DD} - 1.4$	V	
	$I_{VCOMP}$		-0.3		$V_{DD} + 0.3$	V	
Output delay	$t_d$	$V_{DD} = 3.0\text{ V}$ Input slew rate $> 50\text{ mV}/\mu\text{s}$	Comparator high-speed mode, standard mode			1.2	$\mu\text{s}$
			Comparator high-speed mode, window mode			2.0	$\mu\text{s}$
			Comparator low-speed mode, standard mode		3.0	5.0	$\mu\text{s}$
High-electric-potential reference voltage	$V_{TW+}$	Comparator high-speed mode, window mode	$0.66V_{DD}$	$0.76V_{DD}$	$0.86V_{DD}$	V	
Low-electric-potential reference voltage	$V_{TW-}$	Comparator high-speed mode, window mode	$0.14V_{DD}$	$0.24V_{DD}$	$0.34V_{DD}$	V	
Operation stabilization wait time	$t_{CMP}$		100			$\mu\text{s}$	
Internal reference output voltage <sup>Note</sup>	$V_{BGR}$	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , HS (high-speed main) mode	1.38	1.45	1.50	V	

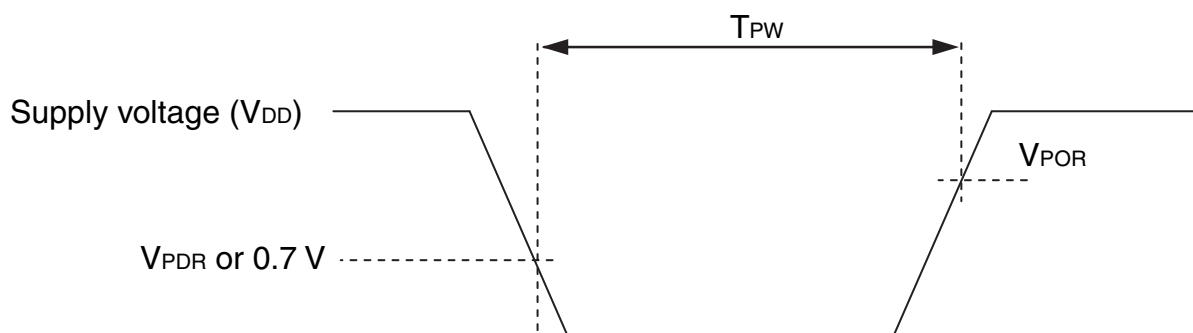
**Note** Cannot be used in LS (low-speed main) mode, LV (low-voltage main) mode, subsystem clock operation, and STOP mode.

## 32.6.4 POR circuit characteristics

 $(T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{SS} = 0$  V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	$V_{POR}$	When power supply rises	1.47	1.51	1.55	V
	$V_{PDR}$	When power supply falls	1.46	1.50	1.54	V
Minimum pulse width <sup>Note</sup>	$T_{PW}$		300			$\mu\text{s}$

**Note** This is the time required for the POR circuit to execute a reset operation when  $V_{DD}$  falls below  $V_{PDR}$ . When the microcontroller enters STOP mode and when the main system clock ( $f_{MAIN}$ ) has been stopped by setting bit 0 (HIOSTOP) and bit 7 (MSTOP) of the clock operation status control register (CSC), this is the time required for the POR circuit to execute a reset operation between when  $V_{DD}$  falls below 0.7 V and when  $V_{DD}$  rises to  $V_{POR}$  or higher.



## 32.6.5 LVD circuit characteristics

## LVD Detection Voltage of Reset Mode and Interrupt Mode

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{PDR} \leq V_{DD} \leq 5.5$  V,  $V_{SS} = 0$  V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	$V_{LVD0}$	When power supply rises	3.98	4.06	4.14	V
			When power supply falls	3.90	3.98	4.06	V
		$V_{LVD1}$	When power supply rises	3.68	3.75	3.82	V
			When power supply falls	3.60	3.67	3.74	V
		$V_{LVD2}$	When power supply rises	3.07	3.13	3.19	V
			When power supply falls	3.00	3.06	3.12	V
		$V_{LVD3}$	When power supply rises	2.96	3.02	3.08	V
			When power supply falls	2.90	2.96	3.02	V
		$V_{LVD4}$	When power supply rises	2.86	2.92	2.97	V
			When power supply falls	2.80	2.86	2.91	V
		$V_{LVD5}$	When power supply rises	2.76	2.81	2.87	V
			When power supply falls	2.70	2.75	2.81	V
		$V_{LVD6}$	When power supply rises	2.66	2.71	2.76	V
			When power supply falls	2.60	2.65	2.70	V
		$V_{LVD7}$	When power supply rises	2.56	2.61	2.66	V
			When power supply falls	2.50	2.55	2.60	V
		$V_{LVD8}$	When power supply rises	2.45	2.50	2.55	V
			When power supply falls	2.40	2.45	2.50	V
		$V_{LVD9}$	When power supply rises	2.05	2.09	2.13	V
			When power supply falls	2.00	2.04	2.08	V
		$V_{LVD10}$	When power supply rises	1.94	1.98	2.02	V
			When power supply falls	1.90	1.94	1.98	V
		$V_{LVD11}$	When power supply rises	1.84	1.88	1.91	V
			When power supply falls	1.80	1.84	1.87	V
$V_{LVD12}$	When power supply rises	1.74	1.77	1.81	V		
	When power supply falls	1.70	1.73	1.77	V		
$V_{LVD13}$	When power supply rises	1.64	1.67	1.70	V		
	When power supply falls	1.60	1.63	1.66	V		
Minimum pulse width	$t_{LW}$		300			$\mu\text{s}$	
Detection delay time					300	$\mu\text{s}$	

**LVD Detection Voltage of Interrupt & Reset Mode**(T<sub>A</sub> = -40 to +85°C, V<sub>PDR</sub> ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Interrupt and reset mode	V <sub>LVD13</sub>	V <sub>POC2</sub> , V <sub>POC1</sub> , V <sub>POC0</sub> = 0, 0, 0, falling reset voltage	1.60	1.63	1.66	V	
	V <sub>LVD12</sub>	LVIS1, LVIS0 = 1, 0	Rising release reset voltage	1.74	1.77	1.81	V
			Falling interrupt voltage	1.70	1.73	1.77	V
	V <sub>LVD11</sub>	LVIS1, LVIS0 = 0, 1	Rising release reset voltage	1.84	1.88	1.91	V
			Falling interrupt voltage	1.80	1.84	1.87	V
	V <sub>LVD4</sub>	LVIS1, LVIS0 = 0, 0	Rising release reset voltage	2.86	2.92	2.97	V
			Falling interrupt voltage	2.80	2.86	2.91	V
	V <sub>LVD11</sub>	V <sub>POC2</sub> , V <sub>POC1</sub> , V <sub>POC0</sub> = 0, 0, 1, falling reset voltage	1.80	1.84	1.87	V	
	V <sub>LVD10</sub>	LVIS1, LVIS0 = 1, 0	Rising release reset voltage	1.94	1.98	2.02	V
			Falling interrupt voltage	1.90	1.94	1.98	V
	V <sub>LVD9</sub>	LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.05	2.09	2.13	V
			Falling interrupt voltage	2.00	2.04	2.08	V
	V <sub>LVD2</sub>	LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.07	3.13	3.19	V
			Falling interrupt voltage	3.00	3.06	3.12	V
	V <sub>LVD8</sub>	V <sub>POC2</sub> , V <sub>POC1</sub> , V <sub>POC0</sub> = 0, 1, 0, falling reset voltage	2.40	2.45	2.50	V	
	V <sub>LVD7</sub>	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 <sub>LVD6</sub>	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 <sub>LVD1</sub>	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 <sub>LVD5</sub>	V <sub>POC2</sub> , V <sub>POC1</sub> , V <sub>POC0</sub> = 0, 1, 1, falling reset voltage	2.70	2.75	2.81	V	
	V <sub>LVD4</sub>	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 <sub>LVD3</sub>	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 <sub>LVD0</sub>	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	

**32.6.6 Supply voltage rising slope characteristics**(T<sub>A</sub> = -40 to +85°C, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V <sub>DD</sub> rising slope	SV <sub>DD</sub>				54	V/ms

**Caution** Make sure to keep the internal reset state by the LVD circuit or an external reset until V<sub>DD</sub> reaches the operating voltage range shown in 32.4 AC Characteristics.

## 32.7 LCD Characteristics

### 32.7.1 External resistance division method

#### (1) Static display mode

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{L4} (\text{MIN.}) \leq V_{DD} \leq 5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	$V_{L4}$		2.0		$V_{DD}$	V

#### (2) 1/2 bias method, 1/4 bias method

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{L4} (\text{MIN.}) \leq V_{DD} \leq 5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	$V_{L4}$		2.7		$V_{DD}$	V

#### (3) 1/3 bias method

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{L4} (\text{MIN.}) \leq V_{DD} \leq 5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	$V_{L4}$		2.5		$V_{DD}$	V



## 32.7.2 Internal voltage boosting method

## (1) 1/3 bias method

(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
LCD output voltage variation range	V <sub>L1</sub>	C1 to C4 <sup>Note 1</sup> = 0.47 μF <sup>Note 2</sup>	VLCD = 04H	0.90	1.00	1.08	V
			VLCD = 05H	0.95	1.05	1.13	V
			VLCD = 06H	1.00	1.10	1.18	V
			VLCD = 07H	1.05	1.15	1.23	V
			VLCD = 08H	1.10	1.20	1.28	V
			VLCD = 09H	1.15	1.25	1.33	V
			VLCD = 0AH	1.20	1.30	1.38	V
			VLCD = 0BH	1.25	1.35	1.43	V
			VLCD = 0CH	1.30	1.40	1.48	V
			VLCD = 0DH	1.35	1.45	1.53	V
			VLCD = 0EH	1.40	1.50	1.58	V
			VLCD = 0FH	1.45	1.55	1.63	V
			VLCD = 10H	1.50	1.60	1.68	V
			VLCD = 11H	1.55	1.65	1.73	V
VLCD = 12H	1.60	1.70	1.78	V			
VLCD = 13H	1.65	1.75	1.83	V			
Doubler output voltage	V <sub>L2</sub>	C1 to C4 <sup>Note 1</sup> = 0.47 μF	2 V <sub>L1</sub> - 0.10	2 V <sub>L1</sub>	2 V <sub>L1</sub>	V	
Tripler output voltage	V <sub>L4</sub>	C1 to C4 <sup>Note 1</sup> = 0.47 μF	3 V <sub>L1</sub> - 0.15	3 V <sub>L1</sub>	3 V <sub>L1</sub>	V	
Reference voltage setup time <sup>Note 2</sup>	t <sub>WAIT1</sub>		5			ms	
Voltage boost wait time <sup>Note 3</sup>	t <sub>WAIT2</sub>	C1 to C4 <sup>Note 1</sup> = 0.47 μF	500			ms	

**Notes** 1. This is a capacitor that is connected between voltage pins used to drive the LCD.

C1: A capacitor connected between CAPH and CAPL

C2: A capacitor connected between V<sub>L1</sub> and GND

C3: A capacitor connected between V<sub>L2</sub> and GND

C4: A capacitor connected between V<sub>L4</sub> and GND

C1 = C2 = C3 = C4 = 0.47 μF ± 30 %

- This is the time required to wait from when the reference voltage is specified by using the VLCD register (or when the internal voltage boosting method is selected (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default value reference voltage is used) until voltage boosting starts (VLCON = 1).
- This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

**(2) 1/4 bias method****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
LCD output voltage variation range	$V_{L1}$	C1 to C5 <sup>Note 1</sup> = $0.47\ \mu\text{F}$ <sup>Note 2</sup>	VLCD = 04H	0.90	1.00	1.08	V
			VLCD = 05H	0.95	1.05	1.13	V
			VLCD = 06H	1.00	1.10	1.18	V
			VLCD = 07H	1.05	1.15	1.23	V
			VLCD = 08H	1.10	1.20	1.28	V
			VLCD = 09H	1.15	1.25	1.33	V
			VLCD = 0AH	1.20	1.30	1.38	V
Doubler output voltage	$V_{L2}$	C1 to C5 <sup>Note 1</sup> = $0.47\ \mu\text{F}$	$2 V_{L1} - 0.08$	$2 V_{L1}$	$2 V_{L1}$	V	
Tripler output voltage	$V_{L3}$	C1 to C5 <sup>Note 1</sup> = $0.47\ \mu\text{F}$	$3 V_{L1} - 0.12$	$3 V_{L1}$	$3 V_{L1}$	V	
Quadruply output voltage	$V_{L4}$	C1 to C5 <sup>Note 1</sup> = $0.47\ \mu\text{F}$	$4 V_{L1} - 0.16$	$4 V_{L1}$	$4 V_{L1}$	V	
Reference voltage setup time <sup>Note 2</sup>	$t_{\text{WAIT1}}$		5			ms	
Voltage boost wait time <sup>Note 3</sup>	$t_{\text{WAIT2}}$	C1 to C5 <sup>Note 1</sup> = $0.47\ \mu\text{F}$	500			ms	

**Notes** 1. This is a capacitor that is connected between voltage pins used to drive the LCD.

C1: A capacitor connected between CAPH and CAPL

C2: A capacitor connected between  $V_{L1}$  and GND

C3: A capacitor connected between  $V_{L2}$  and GND

C4: A capacitor connected between  $V_{L3}$  and GND

C5: A capacitor connected between  $V_{L4}$  and GND

$C1 = C2 = C3 = C4 = C5 = 0.47\ \mu\text{F} \pm 30\%$

- This is the time required to wait from when the reference voltage is specified by using the VLCD register (or when the internal voltage boosting method is selected (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default value reference voltage is used) until voltage boosting starts (VLCON = 1).
- This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

### 32.7.3 Capacitor split method

#### (1) 1/3 bias method

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.2\text{ V} \leq V_D \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$V_{L4}$ voltage	$V_{L4}$	C1 to C4 = $0.47\ \mu\text{F}$ <sup>Note 2</sup>		$V_{DD}$		V
$V_{L2}$ voltage	$V_{L2}$	C1 to C4 = $0.47\ \mu\text{F}$ <sup>Note 2</sup>	$\frac{2}{3} V_{L4} - 0.1$	$\frac{2}{3} V_{L4}$	$\frac{2}{3} V_{L4} + 0.1$	V
$V_{L1}$ voltage	$V_{L1}$	C1 to C4 = $0.47\ \mu\text{F}$ <sup>Note 2</sup>	$\frac{1}{3} V_{L4} - 0.1$	$\frac{1}{3} V_{L4}$	$\frac{1}{3} V_{L4} + 0.1$	V
Capacitor split wait time <sup>Note 1</sup>	$t_{\text{WAIT}}$		100			ms

**Notes** 1. This is the wait time from when voltage bucking is started ( $V_{LCON} = 1$ ) until display is enabled ( $LCDON = 1$ ).

2. This is a capacitor that is connected between voltage pins used to drive the LCD.

C1: A capacitor connected between CAPH and CAPL

C2: A capacitor connected between  $V_{L1}$  and GND

C3: A capacitor connected between  $V_{L2}$  and GND

C4: A capacitor connected between  $V_{L4}$  and GND

$C1 = C2 = C3 = C4 = 0.47\ \mu\text{F} \pm 30\%$

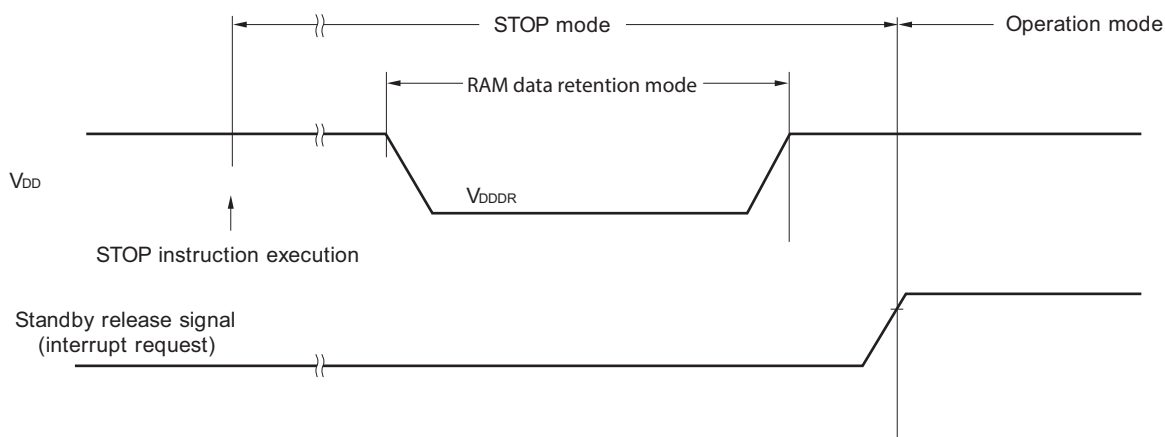
<R> **32.8 RAM Data Retention Characteristics**

(T<sub>A</sub> = -40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V <sub>DDDR</sub>		1.46 <sup>Note</sup>		5.5	V

<R> **Note** This depends on the POR detection voltage. For a falling voltage, data in RAM are retained until the voltage reaches the level that triggers a POR reset but not once it reaches the level at which a POR reset is generated.

<R> **Caution** Data in RAM are not retained if the CPU operates outside the specified operating voltage range. Therefore, place the CPU in STOP mode before the operating voltage drops below the specified range.



**32.9 Flash Memory Programming Characteristics**

(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency	f <sub>CLK</sub>	1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V	1		24	MHz
Number of code flash rewrites <sup>Notes 1, 2, 3</sup>	C <sub>erwr</sub>	Retained for 20 years T <sub>A</sub> = 85°C	1,000			Times
Number of data flash rewrites <sup>Notes 1, 2, 3</sup>		Retained for 1 year T <sub>A</sub> = 25°C		1,000,000		
		Retained for 5 years T <sub>A</sub> = 85°C	100,000			
		Retained for 20 years T <sub>A</sub> = 85°C	10,000			

- Notes**
- 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. This characteristic indicates the flash memory characteristic and based on Renesas Electronics reliability test.

**Remark** When updating data multiple times, use the flash memory as one for updating data.

**32.10 Dedicated Flash Memory Programmer Communication (UART)**

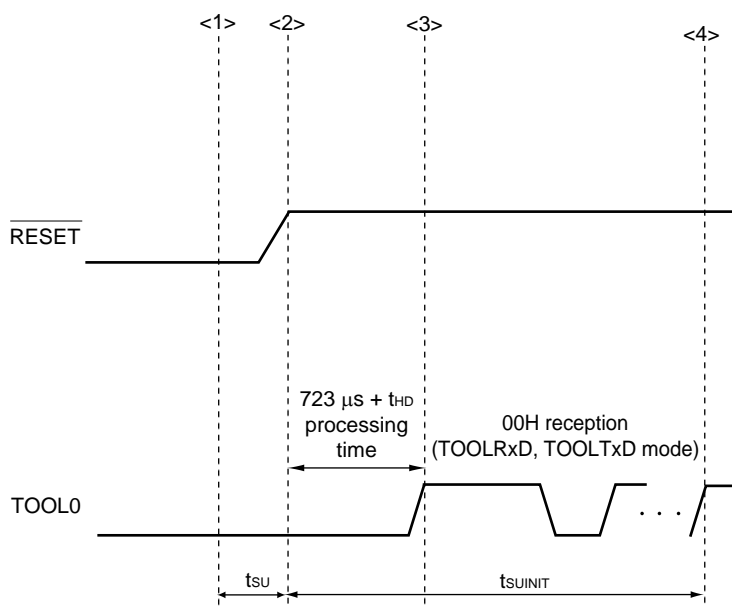
(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

32.11 Timing Specifications for Switching Flash Memory Programming Modes

(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Time to complete the communication for the initial setting after the external reset is released	t <sub>SUINIT</sub>	POR and LVD reset must be released before the external reset is released.			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	t <sub>SU</sub>	POR and LVD reset must be released before the external reset is released.	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)	t <sub>HD</sub>	POR and LVD reset must be released before the external reset is released.	1			ms



- <1> The low level is input to the TOOL0 pin.
- <2> The external reset is released (POR and LVD reset must be released before the external reset is released.).
- <3> The TOOL0 pin is set to the high level.
- <4> Setting of the flash memory programming mode by UART reception and completion the baud rate setting.

**Remark** t<sub>SUINIT</sub>: Communication for the initial setting must be completed within 100 ms after the external reset is released during this period.

t<sub>SU</sub>: Time to release the external reset after the TOOL0 pin is set to the low level

t<sub>HD</sub>: Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)

**CHAPTER 33 ELECTRICAL SPECIFICATIONS (G: INDUSTRIAL APPLICATIONS  $T_A = -40$  to  $+105^\circ\text{C}$ )**

This chapter describes the following electrical specifications.

Target products G: Industrial applications  $T_A = -40$  to  $+105^\circ\text{C}$

R5F10WLAGFB, R5F10WLCGFB, R5F10WLDGFB,  
R5F10WLEGFB, R5F10WLFGFB, R5F10WLGGFB  
R5F10WMAGFB, R5F10WMCGB, R5F10WMDGFB,  
R5F10WMEGFB, R5F10WMFGFB, R5F10WMGGFB

- Cautions**
1. The RL78/L13 microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.
  2. The pins mounted depend on the product. See 2.1 Port Function to 2.2.1 With functions for each product.
  3. Consult Renesas salesperson and distributor for derating when the product is used at  $T_A = +85^\circ\text{C}$  to  $+105^\circ\text{C}$ . Note that derating means “systematically lowering the load from the rated value to improve reliability”.

<R> **Remark** When RL78/L13 is used in the range of  $T_A = -40$  to  $+85^\circ\text{C}$ , see **CHAPTER 32 ELECTRICAL SPECIFICATIONS ( $T_A = -40$  to  $+85^\circ\text{C}$ )**.

“G: Industrial applications ( $T_A = -40$  to  $+105^\circ\text{C}$ ) differ from “A: Consumer applications” in function as follows:

Fields of Application	A: Consumer applications	G: Industrial applications
Operating ambient temperature	$T_A = -40$ to $+85^\circ\text{C}$	$T_A = -40$ to $+105^\circ\text{C}$
Operation mode operating voltage range	HS (high-speed main) mode: $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz}$ to $24\text{ MHz}$ $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz}$ to $16\text{ MHz}$ LS (low-speed main) mode: $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz}$ to $8\text{ MHz}$ LV (low-voltage main) mode: $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz}$ to $4\text{ MHz}$	HS (high-speed main) mode only: $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz}$ to $24\text{ MHz}$ $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz}$ to $16\text{ MHz}$
High-speed on-chip oscillator clock accuracy	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ : $\pm 1.0\%$ @ $T_A = -20$ to $+85^\circ\text{C}$ $\pm 1.5\%$ @ $T_A = -40$ to $-20^\circ\text{C}$ $1.6\text{ V} \leq V_{DD} < 1.8\text{ V}$ : $\pm 5.0\%$ @ $T_A = -20$ to $+85^\circ\text{C}$ $\pm 5.5\%$ @ $T_A = -40$ to $-20^\circ\text{C}$	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ : $\pm 2.0\%$ @ $T_A = +85$ to $+105^\circ\text{C}$ $\pm 1.0\%$ @ $T_A = -20$ to $+85^\circ\text{C}$ $\pm 1.5\%$ @ $T_A = -40$ to $-20^\circ\text{C}$
Serial array unit	UART CSI: $f_{CLK}/2$ (16 Mbps supported), $f_{CLK}/4$ Simplified I <sup>2</sup> C	UART CSI: $f_{CLK}/4$ Simplified I <sup>2</sup> C
IICA	Standard mode Fast mode Fast mode plus	Standard mode Fase mode
Voltage detector	<ul style="list-style-type: none"> <li>Rising: 1.67 V to 4.06 V (14 levels)</li> <li>Falling: 1.63 V to 3.98 V (14 levels)</li> </ul>	<ul style="list-style-type: none"> <li>Rising: 2.61 V to 4.06 V (8 levels)</li> <li>Falling: 2.55 V to 3.98 V (8 levels)</li> </ul>

**Remark** Electrical specifications of G: Industrial applications ( $T_A = -40$  to  $+105^\circ\text{C}$ ) differ from “A: Consumer applications”. For details, see 33.1 to 33.11 below.

### 33.1 Absolute Maximum Ratings

#### Absolute Maximum Ratings (1/3)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V <sub>DD</sub>		-0.5 to +6.5	V
REGC pin input voltage	V <sub>IREGC</sub>	REGC	-0.3 to +2.8 and -0.3 to V <sub>DD</sub> +0.3 <sup>Note 1</sup>	V
Input voltage	V <sub>I1</sub>	P00 to P07, P10 to P17, P20 to P27, P30 to P35, P40 to P47, P50 to P57, P60, P61, P70 to P77, P121 to P127, P130, P137	-0.3 to V <sub>DD</sub> +0.3 <sup>Note 2</sup>	V
	V <sub>I2</sub>	P60 and P61 (N-ch open-drain)	-0.3 to +6.5	V
	V <sub>I3</sub>	EXCLK, EXCLKS, RESET	-0.3 to V <sub>DD</sub> +0.3 <sup>Note 2</sup>	V
Output voltage	V <sub>O1</sub>	P00 to P07, P10 to P17, P20 to P27, P30 to P35, P40 to P47, P50 to P57, P60, P61, P70 to P77, P121 to P127, P130, P137	-0.3 to V <sub>DD</sub> +0.3 <sup>Note 2</sup>	V
Analog input voltage	V <sub>AI1</sub>	ANI0, ANI1, ANI16 to ANI26	-0.3 to V <sub>DD</sub> +0.3 and -0.3 to AV <sub>REF(+)</sub> +0.3 <sup>Notes 2, 3</sup>	V

**Notes 1.** Connect the REGC pin to V<sub>SS</sub> 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<sub>REF(+)</sub> + 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<sub>REF(+)</sub>: + side reference voltage of the A/D converter.

**3.** V<sub>SS</sub>: Reference voltage



**Absolute Maximum Ratings (2/3)**

Parameter	Symbol	Conditions	Ratings	Unit	
LCD voltage	V <sub>L1</sub>	V <sub>L1</sub> voltage <sup>Note 1</sup>	-0.3 to +2.8 and -0.3 to V <sub>L4</sub> +0.3	V	
	V <sub>L2</sub>	V <sub>L2</sub> voltage <sup>Note 1</sup>	-0.3 to V <sub>L4</sub> +0.3 <sup>Note 2</sup>	V	
	V <sub>L3</sub>	V <sub>L3</sub> voltage <sup>Note 1</sup>	-0.3 to V <sub>L4</sub> +0.3 <sup>Note 2</sup>	V	
	V <sub>L4</sub>	V <sub>L4</sub> voltage <sup>Note 1</sup>	-0.3 to +6.5	V	
	V <sub>LCAP</sub>	CAPL, CAPH voltage <sup>Note 1</sup>	-0.3 to V <sub>L4</sub> +0.3 <sup>Note 2</sup>	V	
	V <sub>OUT</sub>	COM0 to COM7 SEG0 to SEG50 output voltage	External resistance division method	-0.3 to V <sub>DD</sub> +0.3 <sup>Note 2</sup>	V
			Capacitor split method	-0.3 to V <sub>DD</sub> +0.3 <sup>Note 2</sup>	V
Internal voltage boosting method			-0.3 to V <sub>L4</sub> +0.3 <sup>Note 2</sup>	V	

**Notes 1.** This value only indicates the absolute maximum ratings when applying voltage to the V<sub>L1</sub>, V<sub>L2</sub>, V<sub>L3</sub>, and V<sub>L4</sub> pins; it does not mean that applying voltage to these pins is recommended. When using the internal voltage boosting method or capacitance split method, connect these pins to V<sub>SS</sub> via a capacitor (0.47 μF ± 30%) and connect a capacitor (0.47 μF ± 30%) between the CAPL and CAPH pins.

**2.** Must be 6.5 V or lower.

**Caution** Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

**Remark** V<sub>SS</sub>: Reference voltage

**Absolute Maximum Ratings (T<sub>A</sub> = 25°C) (3/3)**

Parameter	Symbol	Conditions		Ratings	Unit
<R> Output current, high	I <sub>OH1</sub>	Per pin	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P60, P61, P70 to P77, P125 to P127, P130	-40	mA
<R>		Total of all pins -170 mA	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P60, P61, P70 to P77, P125 to P127, P130	-170	mA
<R>	I <sub>OH2</sub>	Per pin	P20, P21	-0.5	mA
<R>		Total of all pins		-1	mA
<R> Output current, low	I <sub>OL1</sub>	Per pin	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P60, P61, P70 to P77, P125 to P127, P130	40	mA
<R>		Total of all pins 170 mA	P40 to P47, P130	70	mA
<R>	I <sub>OL2</sub>		P00 to P07, P10 to P17, P22 to P27, P30 to P35, P50 to P57, P60, P61, P70 to P77, P125 to P127	100	mA
<R>		Per pin	P20, P21	1	mA
<R>		Total of all pins		2	mA
Operating ambient temperature	T <sub>A</sub>	In normal operation mode		-40 to +105	°C
		In flash memory programming mode			°C
Storage temperature	T <sub>stg</sub>			-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.

## 33.2 Oscillator Characteristics

### 33.2.1 X1 and XT1 oscillator characteristics

( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency ( $f_x$ ) <sup>Note</sup>	Ceramic resonator/ crystal resonator	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	1.0		20.0	MHz
		$2.4\text{ V} \leq V_{DD} < 2.7\text{ V}$	1.0		16.0	
XT1 clock oscillation frequency ( $f_{XT}$ ) <sup>Note</sup>	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, see **5.4 System Clock Oscillator**.

**33.2.2 On-chip oscillator characteristics****( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency <small>Notes 1, 2</small>	$f_{IH}$			1		24	MHz
High-speed on-chip oscillator clock frequency accuracy		+85 to $+105^\circ\text{C}$	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	-2		+2	%
		-20 to $+85^\circ\text{C}$	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	-1		+1	%
		-40 to $-20^\circ\text{C}$	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	-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.** The high-speed on-chip oscillator frequency is selected by bits 0 to 4 of the option byte (000C2H/010C2H) and bits 0 to 2 of the HOCODIV register.

**2.** This indicates the oscillator characteristics only. Refer to **AC Characteristics** for the instruction execution time.

### 33.3 DC Characteristics

#### 33.3.1 Pin characteristics

(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Output current, high <sup>Note 1</sup>	I <sub>OH1</sub>	Per pin for P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V			-3.0 <sup>Note 2</sup>	mA
		Total of P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130 (When duty = 70% <sup>Note 3</sup> )	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V			-45.0	mA
			2.7 V ≤ V <sub>DD</sub> < 4.0 V			-15.0	mA
			2.4 V ≤ V <sub>DD</sub> < 2.7 V			-7.0	mA
	I <sub>OH2</sub>	Per pin for P20 and P21	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V			-0.1 <sup>Note 2</sup>	mA
		Total of all pins (When duty = 70% <sup>Note 3</sup> )	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V			-0.2	mA

- Notes**
- Value of the current at which the device operation is guaranteed even if the current flows from the V<sub>DD</sub> pin to an output pin
  - Do not exceed the total current value.
  - Output current value 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 factor from 70% to n%).

- Total output current of pins = (I<sub>OH</sub> × 0.7)/(n × 0.01)

<Example> Where n = 80% and I<sub>OH</sub> = -45.0 mA

$$\text{Total output current of pins} = (-45.0 \times 0.7)/(80 \times 0.01) = -39.375 \text{ mA}$$

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, P04 to P07, P16, P17, P35, P42 to P44, P46, P47, P53 to P56, and P130 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<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output current, I <sub>OL</sub> <sup>Note 1</sup>	I <sub>OL1</sub>	Per pin for P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130			8.5 <sup>Note 2</sup>	mA
		Per pin for P60 and P61			15.0 <sup>Note 2</sup>	mA
		Total of P40 to P47, P130 (When duty = 70% <sup>Note 3</sup> )	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V		40.0	mA
			2.7 V ≤ V <sub>DD</sub> < 4.0 V		15.0	mA
			2.4 V ≤ V <sub>DD</sub> < 2.7 V		9.0	mA
		Total of P00 to P07, P10 to P17, P22 to P27, P30 to P35, P50 to P57, P70 to P77, P125 to P127 (When duty = 70% <sup>Note 3</sup> )	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V		60.0	mA
			2.7 V ≤ V <sub>DD</sub> < 4.0 V		35.0	mA
	2.4 V ≤ V <sub>DD</sub> < 2.7 V			20.0	mA	
	Total of all pins (When duty = 70% <sup>Note 3</sup> )			100.0	mA	
	I <sub>OL2</sub>	Per pin for P20 and P21			0.4 <sup>Note 2</sup>	mA
Total of all pins (When duty = 70% <sup>Note 3</sup> )		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		0.8	mA	

**Notes** 1. Value of the current at which the device operation is guaranteed even if the current flows from an output pin to the V<sub>SS</sub> pin

2. Do not exceed the total current value.

3. Output current value 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 factor from 70% to n%).

- Total output current of pins = (I<sub>OL</sub> × 0.7)/(n × 0.01)

<Example> Where n = 80% and I<sub>OL</sub> = 40.0 mA

$$\text{Total output current of pins} = (40.0 \times 0.7)/(80 \times 0.01) = 35.0 \text{ mA}$$

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  $+105^\circ\text{C}$ ,  $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input voltage, high	$V_{IH1}$	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130, P137	Normal input buffer	$0.8V_{DD}$		$V_{DD}$	V
	$V_{IH2}$	P03, P05, P06, P16, P17, P34, P43, P44, P46, P47, P53, P55	TTL input buffer $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.2		$V_{DD}$	V
			TTL input buffer $3.3\text{ V} \leq V_{DD} < 4.0\text{ V}$	2.0		$V_{DD}$	V
			TTL input buffer $2.4\text{ V} \leq V_{DD} < 3.3\text{ V}$	1.5		$V_{DD}$	V
	$V_{IH3}$	P20, P21		$0.7V_{DD}$		$V_{DD}$	V
	$V_{IH4}$	P60, P61		$0.7V_{DD}$		6.0	V
	$V_{IH5}$	P121 to P124, P137, EXCLK, EXCLKS, RESET		$0.8V_{DD}$		$V_{DD}$	V
Input voltage, low	$V_{IL1}$	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130, P137	Normal input buffer	0		$0.2V_{DD}$	V
	$V_{IL2}$	P03, P05, P06, P16, P17, P34, P43, P44, P46, P47, P53, P55	TTL input buffer $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0		0.8	V
			TTL input buffer $3.3\text{ V} \leq V_{DD} < 4.0\text{ V}$	0		0.5	V
			TTL input buffer $2.4\text{ V} \leq V_{DD} < 3.3\text{ V}$	0		0.32	V
	$V_{IL3}$	P20, P21		0		$0.3V_{DD}$	V
	$V_{IL4}$	P60, P61		0		$0.3V_{DD}$	V
	$V_{IL5}$	P121 to P124, P137, EXCLK, EXCLKS, RESET		0		$0.2V_{DD}$	V

**Caution** The maximum value of  $V_{IH}$  of pins P00, P04 to P07, P16, P17, P35, P42 to P44, P46, P47, P53 to P56, and P130 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  $+105^\circ\text{C}$ ,  $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output voltage, high	V <sub>OH1</sub>	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130	4.0 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OH1</sub> = -3.0 mA	V <sub>DD</sub> - 0.7		V
			2.7 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OH1</sub> = -2.0 mA	V <sub>DD</sub> - 0.6		V
			2.4 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OH1</sub> = -1.5 mA	V <sub>DD</sub> - 0.5		V
	V <sub>OH2</sub>	P20 and P21	2.4 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OH2</sub> = -100 $\mu$ A	V <sub>DD</sub> - 0.5		V
Output voltage, low	V <sub>OL1</sub>	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130	4.0 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OL1</sub> = 8.5 mA		0.7	V
			2.7 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OL1</sub> = 3.0 mA		0.6	V
			2.7 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OL1</sub> = 1.5 mA		0.4	V
			2.4 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OL1</sub> = 0.6 mA		0.4	V
	V <sub>OL2</sub>	P20 and P21	2.4 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OL2</sub> = 400 $\mu$ A		0.4	V
	V <sub>OL3</sub>	P60 and P61	4.0 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OL3</sub> = 15.0 mA		2.0	V
			4.0 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OL3</sub> = 5.0 mA		0.4	V
			2.7 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OL3</sub> = 3.0 mA		0.4	V
			2.4 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, I <sub>OL3</sub> = 2.0 mA		0.4	V

**Caution** P00, P04 to P07, P16, P17, P35, P42 to P44, P46, P47, P53 to P56, and P130 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  $+105^\circ\text{C}$ ,  $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit		
Input leakage current, high	I <sub>LH1</sub>	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130, P137	V <sub>I</sub> = V <sub>DD</sub>		1	μA		
	I <sub>LH2</sub>	P20 and P21, $\overline{\text{RESET}}$	V <sub>I</sub> = V <sub>DD</sub>		1	μA		
	I <sub>LH3</sub>	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	V <sub>I</sub> = V <sub>DD</sub>		1	μA		
			In input port mode and when external clock is input					
		Resonator connected		10	μA			
Input leakage current, low	I <sub>LIL1</sub>	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P70 to P77, P125 to P127, P130, P137	V <sub>I</sub> = V <sub>SS</sub>		-1	μA		
	I <sub>LIL2</sub>	P20 and P21, $\overline{\text{RESET}}$	V <sub>I</sub> = V <sub>SS</sub>		-1	μA		
	I <sub>LIL3</sub>	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	V <sub>I</sub> = V <sub>SS</sub>		-1	μA		
			In input port mode and when external clock is input					
		Resonator connected		-10	μA			
On-chip pull-up resistance	R <sub>U1</sub>	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P45 to P47, P50 to P57, P70 to P77, P125 to P127, P130	V <sub>I</sub> = V <sub>SS</sub>		10	20	100	kΩ
	R <sub>U2</sub>	P40 to P44	V <sub>I</sub> = V <sub>SS</sub>		10	20	100	kΩ

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

## 33.3.2 Supply current characteristics

(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

(1/2)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit	
Supply current	I <sub>DD1</sub> <sup>Note 1</sup>	Operating mode	HS (high-speed main) mode <sup>Note 5</sup>	f <sub>HOCO</sub> = 48 MHz <sup>Note 3</sup> , f <sub>IH</sub> = 24 MHz <sup>Note 3</sup>	Basic operation	V <sub>DD</sub> = 5.0 V		2.0		mA
						V <sub>DD</sub> = 3.0 V		2.0		mA
				Normal operation	V <sub>DD</sub> = 5.0 V		3.8	7.0	mA	
					V <sub>DD</sub> = 3.0 V		3.8	7.0	mA	
				Basic operation	V <sub>DD</sub> = 5.0 V		1.7		mA	
					V <sub>DD</sub> = 3.0 V		1.7		mA	
			Normal operation	V <sub>DD</sub> = 5.0 V		3.6	6.5	mA		
				V <sub>DD</sub> = 3.0 V		3.6	6.5	mA		
			Normal operation	V <sub>DD</sub> = 5.0 V		2.7	5.0	mA		
				V <sub>DD</sub> = 3.0 V		2.7	5.0	mA		
			HS (high-speed main) mode <sup>Note 5</sup>	f <sub>MX</sub> = 20 MHz <sup>Note 2</sup> , V <sub>DD</sub> = 5.0 V	Normal operation	Square wave input		3.0	5.4	mA
						Resonator connection		3.2	5.6	mA
		f <sub>MX</sub> = 20 MHz <sup>Note 2</sup> , V <sub>DD</sub> = 3.0 V		Normal operation	Square wave input		2.9	5.4	mA	
					Resonator connection		3.2	5.6	mA	
		f <sub>MX</sub> = 10 MHz <sup>Note 2</sup> , V <sub>DD</sub> = 5.0 V		Normal operation	Square wave input		1.9	3.2	mA	
					Resonator connection		1.9	3.2	mA	
		f <sub>MX</sub> = 10 MHz <sup>Note 2</sup> , V <sub>DD</sub> = 3.0 V		Normal operation	Square wave input		1.9	3.2	mA	
					Resonator connection		1.9	3.2	mA	
		Subsystem clock operation	f <sub>SUB</sub> = 32.768 kHz <sup>Note 4</sup> , T <sub>A</sub> = -40°C	Normal operation	Square wave input		4.0	5.4	μA	
					Resonator connection		4.3	5.4	μA	
			f <sub>SUB</sub> = 32.768 kHz <sup>Note 4</sup> , T <sub>A</sub> = +25°C	Normal operation	Square wave input		4.0	5.4	μA	
					Resonator connection		4.3	5.4	μA	
			f <sub>SUB</sub> = 32.768 kHz <sup>Note 4</sup> , T <sub>A</sub> = +50°C	Normal operation	Square wave input		4.1	7.1	μA	
					Resonator connection		4.4	7.1	μA	
f <sub>SUB</sub> = 32.768 kHz <sup>Note 4</sup> , T <sub>A</sub> = +70°C	Normal operation		Square wave input		4.3	8.7	μA			
			Resonator connection		4.7	8.7	μA			
f <sub>SUB</sub> = 32.768 kHz <sup>Note 4</sup> , T <sub>A</sub> = +85°C	Normal operation		Square wave input		4.7	12.0	μA			
			Resonator connection		5.2	12.0	μA			
f <sub>SUB</sub> = 32.768 kHz <sup>Note 4</sup> , T <sub>A</sub> = +105°C	Normal operation		Square wave input		6.4	35.0	μA			
			Resonator connection		6.6	35.0	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 values below the MAX. column include the peripheral operation current. However, not including the current flowing into the LCD controller/driver, A/D converter, LVD circuit, comparator, I/O port, on-chip pull-up/pull-down resistors, and the current flowing during data flash rewrite.
  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 setting ultra-low power consumption oscillation ( $AMP_{HS1} = 1$ ). The current flowing into the LCD controller/driver, 16-bit timer KB20, real-time clock 2, 12-bit interval timer, and watchdog timer is not included.
  5. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.  
HS (high-speed main) mode:  $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz to }24\text{ MHz}$   
 $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz to }16\text{ MHz}$

- Remarks**
1.  $f_{MX}$ : High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)
  2.  $f_{HOCO}$ : High-speed on-chip oscillator clock frequency (48 MHz max.)
  3.  $f_{IH}$ : High-speed on-chip oscillator clock frequency (24 MHz max.)
  4.  $f_{SUB}$ : Subsystem clock frequency (XT1 clock oscillation frequency)
  5. Except subsystem clock operation, temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$

(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

(2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit		
Supply current Note 1	I <sub>DD2</sub> <sup>Note 2</sup>	HALT mode	HS (high-speed main) mode <sup>Note 7</sup>	f <sub>HOCO</sub> = 48 MHz <sup>Note 4</sup> , f <sub>IH</sub> = 24 MHz <sup>Note 4</sup>	V <sub>DD</sub> = 5.0 V	0.71	2.55	mA	
					V <sub>DD</sub> = 3.0 V	0.71	2.55	mA	
				f <sub>HOCO</sub> = 24 MHz <sup>Note 4</sup> , f <sub>IH</sub> = 24 MHz <sup>Note 4</sup>	V <sub>DD</sub> = 5.0 V	0.49	1.95	mA	
					V <sub>DD</sub> = 3.0 V	0.49	1.95	mA	
				f <sub>HOCO</sub> = 16 MHz <sup>Note 4</sup> , f <sub>IH</sub> = 16 MHz <sup>Note 4</sup>	V <sub>DD</sub> = 5.0 V	0.43	1.50	mA	
					V <sub>DD</sub> = 3.0 V	0.43	1.50	mA	
				HS (high-speed main) mode <sup>Note 7</sup>	f <sub>MX</sub> = 20 MHz <sup>Note 3</sup> , V <sub>DD</sub> = 5.0 V	Square wave input	0.31	1.76	mA
			Resonator connection			0.48	1.92	mA	
			f <sub>MX</sub> = 20 MHz <sup>Note 3</sup> , V <sub>DD</sub> = 3.0 V		Square wave input	0.29	1.76	mA	
					Resonator connection	0.48	1.92	mA	
			f <sub>MX</sub> = 10 MHz <sup>Note 3</sup> , V <sub>DD</sub> = 5.0 V		Square wave input	0.20	0.96	mA	
					Resonator connection	0.28	1.07	mA	
		Subsystem clock operation	f <sub>SUB</sub> = 32.768 kHz <sup>Note 5</sup> , T <sub>A</sub> = -40°C	Square wave input	0.34	0.62	μA		
				Resonator connection	0.51	0.80	μA		
			f <sub>SUB</sub> = 32.768 kHz <sup>Note 5</sup> , T <sub>A</sub> = +25°C	Square wave input	0.38	0.62	μA		
				Resonator connection	0.57	0.80	μA		
			f <sub>SUB</sub> = 32.768 kHz <sup>Note 5</sup> , T <sub>A</sub> = +50°C	Square wave input	0.46	2.30	μA		
				Resonator connection	0.67	2.49	μA		
	f <sub>SUB</sub> = 32.768 kHz <sup>Note 5</sup> , T <sub>A</sub> = +70°C	Square wave input	0.65	4.03	μA				
		Resonator connection	0.91	4.22	μA				
	f <sub>SUB</sub> = 32.768 kHz <sup>Note 5</sup> , T <sub>A</sub> = +85°C	Square wave input	1.00	8.04	μA				
		Resonator connection	1.31	8.23	μA				
	f <sub>SUB</sub> = 32.768 kHz <sup>Note 5</sup> , T <sub>A</sub> = +105°C	Square wave input	3.05	27.00	μA				
Resonator connection		3.24	27.00	μA					
I <sub>DD3</sub> <sup>Note 6</sup>	STOP mode <sup>Note 8</sup>	T <sub>A</sub> = -40°C			0.18	0.52	μA		
		T <sub>A</sub> = +25°C			0.24	0.52	μA		
		T <sub>A</sub> = +50°C			0.33	2.21	μA		
		T <sub>A</sub> = +70°C			0.53	3.94	μA		
		T <sub>A</sub> = +85°C			0.93	7.95	μA		
		T <sub>A</sub> = +105°C			2.91	25.00	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 values below the MAX. column include the peripheral operation current. However, not including the current flowing into the LCD controller/driver, A/D converter, LVD circuit, comparator, I/O port, on-chip pull-up/pull-down resistors, and the current flowing during data flash rewrite.
  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 ( $AMPHS1 = 1$ ). The current flowing into the real-time clock 2 is included. The current flowing into the clock output/buzzer output, 12-bit interval timer, and watchdog timer is not included.
  6. The current flowing into the real-time clock 2, clock output/buzzer output, 12-bit interval timer, and watchdog timer is not included.
  7. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.  
HS (high-speed main) mode:  $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz to }24\text{ MHz}$   
 $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}@1\text{ MHz to }16\text{ MHz}$
  8. Regarding the value for current to operate the subsystem clock in STOP mode, refer to that in HALT mode.

- Remarks**
1.  $f_{MX}$ : High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)
  2.  $f_{HOCO}$ : High-speed on-chip oscillator clock frequency (48 MHz max.)
  3.  $f_{IH}$ : High-speed on-chip oscillator clock frequency (24 MHz max.)
  4.  $f_{SUB}$ : Subsystem clock frequency (XT1 clock oscillation frequency)
  5. Except subsystem clock operation and STOP mode, temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$

(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit	
Low-speed on-chip oscillator operating current	I <sub>FIL</sub> <sup>Note 1</sup>					0.20		μA	
RTC2 operating current	I <sub>RTC</sub> <sup>Notes 1, 2, 3</sup>	f <sub>SUB</sub> = 32.768 kHz				0.02		μA	
12-bit interval timer operating current	I <sub>TMKA</sub> <sup>Notes 1, 2, 4</sup>					0.04		μA	
Watchdog timer operating current	I <sub>WDT</sub> <sup>Notes 1, 2, 5</sup>	f <sub>IL</sub> = 15 kHz				0.22		μA	
A/D converter operating current	I <sub>ADC</sub> <sup>Notes 1, 6</sup>	When conversion at maximum speed	Normal mode, AV <sub>REFP</sub> = V <sub>DD</sub> = 5.0 V			1.3	1.7	mA	
			Low voltage mode, AV <sub>REFP</sub> = V <sub>DD</sub> = 3.0 V			0.5	0.7	mA	
A/D converter reference voltage current	I <sub>ADREF</sub> <sup>Note 1</sup>					75.0		μA	
Temperature sensor operating current	I <sub>TMPS</sub> <sup>Note 1</sup>					75.0		μA	
LVD operating current	I <sub>LVD</sub> <sup>Notes 1, 7</sup>					0.08		μA	
Comparator operating current	I <sub>COMP</sub> <sup>Notes 1, 11</sup>	V <sub>DD</sub> = 5.0 V, Regulator output voltage = 2.1 V	Window mode			12.5		μA	
			Comparator high-speed mode			6.5		μA	
			Comparator low-speed mode			1.7		μA	
		V <sub>DD</sub> = 5.0 V, Regulator output voltage = 1.8 V	Window mode			8.0		μA	
			Comparator high-speed mode			4.0		μA	
			Comparator low-speed mode			1.3		μA	
Self-programming operating current	I <sub>FSP</sub> <sup>Notes 1, 9</sup>					2.00	12.20	mA	
BGO operating current	I <sub>BGO</sub> <sup>Notes 1, 8</sup>					2.00	12.20	mA	
SNOOZE operating current	I <sub>SNOZ</sub> <sup>Note 1</sup>	ADC operation	While the mode is shifting <sup>Note 10</sup>			0.50	0.60	mA	
			During A/D conversion, in low voltage mode, AV <sub>REFP</sub> = V <sub>DD</sub> = 3.0 V			1.20	1.44	mA	
		CSI/UART operation			0.70	0.84	mA		
LCD operating current	I <sub>LCD1</sub> <sup>Notes 1, 12, 13</sup>	External resistance division method	f <sub>LCD</sub> = f <sub>SUB</sub> LCD clock = 128 Hz	1/3 bias, four time slices	V <sub>DD</sub> = 5.0 V, V <sub>L4</sub> = 5.0 V		0.04	0.20	μA
		Internal voltage boosting method	f <sub>LCD</sub> = f <sub>SUB</sub> LCD clock = 128 Hz	1/3 bias, four time slices	V <sub>DD</sub> = 3.0 V, V <sub>L4</sub> = 3.0 V (V <sub>LCD</sub> = 04H)		0.85	2.20	μA
	V <sub>DD</sub> = 5.0 V, V <sub>L4</sub> = 5.1 V (V <sub>LCD</sub> = 12H)					1.55	3.70	μA	
	I <sub>LCD3</sub> <sup>Note 1, 12</sup>	Capacitor split method	f <sub>LCD</sub> = f <sub>SUB</sub> LCD clock = 128 Hz	1/3 bias, four time slices	V <sub>DD</sub> = 3.0 V, V <sub>L4</sub> = 3.0 V		0.20	0.50	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 2 (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The value of the current for the RL78 microcontrollers is the sum of the values of either  $I_{DD1}$  or  $I_{DD2}$ , and  $I_{RTC}$ , when the real-time clock 2 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 real-time clock 2.
  4. Current flowing only to the 12-bit interval timer (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The value of the current for the RL78 microcontrollers is the sum of the values of either  $I_{DD1}$  or  $I_{DD2}$ , and  $I_{TMKA}$ , 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 current value of the RL78 microcontrollers is the sum of  $I_{DD1}$ ,  $I_{DD2}$  or  $I_{DD3}$  and  $I_{WDT}$  when the watchdog timer operates.
  6. Current flowing only to the A/D converter. The current value 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 microcontrollers is the sum of  $I_{DD1}$ ,  $I_{DD2}$  or  $I_{DD3}$  and  $I_{LVD}$  when the LVD circuit operates.
  8. Current flowing only during data flash rewrite.
  9. Current flowing only during self programming.
  10. For shift time to the SNOOZE mode, see **21.3.3 SNOOZE mode**.
  11. Current flowing only to the comparator circuit. The current value of the RL78 microcontrollers is the sum of  $I_{DD1}$ ,  $I_{DD2}$  or  $I_{DD3}$  and  $I_{CMP}$  when the comparator circuit operates.
  12. Current flowing only to the LCD controller/driver. The value of the current for the RL78 microcontrollers is the sum of the supply current ( $I_{DD1}$  or  $I_{DD2}$ ) and LCD operating current ( $I_{LCD1}$ ,  $I_{LCD2}$ , or  $I_{LCD3}$ ), when the LCD controller/driver operates in operation mode or HALT mode. However, not including the current flowing into the LCD panel. Conditions of the TYP. value and MAX. value are as follows.
    - Setting 20 pins as the segment function and blinking all
    - Selecting  $f_{SUB}$  for system clock when LCD clock = 128 Hz ( $LCDC0 = 07H$ )
    - Setting four time slices and 1/3 bias
  13. Not including the current flowing into the external division resistor when using the external resistance division method.

- 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. The temperature condition for the TYP. value is  $T_A = 25^\circ\text{C}$ .

## 33.4 AC Characteristics

(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	
Instruction cycle (minimum instruction execution time)	T <sub>CY</sub>	Main system clock (f <sub>MAIN</sub> ) operation	HS (high-speed main) mode	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.0417	1	μs	
				2.4 V ≤ V <sub>DD</sub> < 2.7 V	0.0625	1	μs	
		Subsystem clock (f <sub>SUB</sub> ) operation		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V	28.5	30.5	31.3	μs
		In the self programming mode	HS (high-speed main) mode	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.0417	1	μs	
2.4 V ≤ V <sub>DD</sub> < 2.7 V	0.0625			1	μs			
External system clock frequency	f <sub>EX</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V		1.0		20.0	MHz	
		2.4 V ≤ V <sub>DD</sub> < 2.7 V		1.0		16.0	MHz	
	f <sub>EXS</sub>			32		35	kHz	
External system clock input high-level width, low-level width	t <sub>EXH</sub> , t <sub>EXL</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V		24			ns	
		2.4 V ≤ V <sub>DD</sub> < 2.7 V		30			ns	
	t <sub>EXHS</sub> , t <sub>EXLS</sub>			13.7			μs	
TI00 to TI07 input high-level width, low-level width	t <sub>TIH</sub> , t <sub>TIL</sub>			1/f <sub>MCK</sub> +10			ns	
TO00 to TO07, TKBO0 <sup>Note</sup> , TKBO01-0 to TKBO01-2 <sup>Note</sup> output frequency	f <sub>TO</sub>	HS (high-speed main) mode	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V			12	MHz	
			2.7 V ≤ V <sub>DD</sub> < 4.0 V			8	MHz	
			2.4 V ≤ V <sub>DD</sub> < 2.7 V			4	MHz	
PCLBUZ0, PCLBUZ1 output frequency	f <sub>PCL</sub>	HS (high-speed main) mode	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V			16	MHz	
			2.7 V ≤ V <sub>DD</sub> < 4.0 V			8	MHz	
			2.4 V ≤ V <sub>DD</sub> < 2.7 V			4	MHz	
Interrupt input high-level width, low-level width	t <sub>INTH</sub> , t <sub>INTL</sub>	INTP0 to INTP7	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V	1			μs	
Key interrupt input high-level width, low-level width	t <sub>KRH</sub> , t <sub>KRL</sub>	KR0 to KR7	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V	250			ns	
IH-PWM output restart input high-level width	t <sub>IHR</sub>	INTP0 to INTP7		2			f <sub>CLK</sub>	
TMKB2 forced output stop input high-level width	t <sub>IHR</sub>	INTP0 to INTP2		2			f <sub>CLK</sub>	
RESET low-level width	t <sub>RSL</sub>			10			μs	

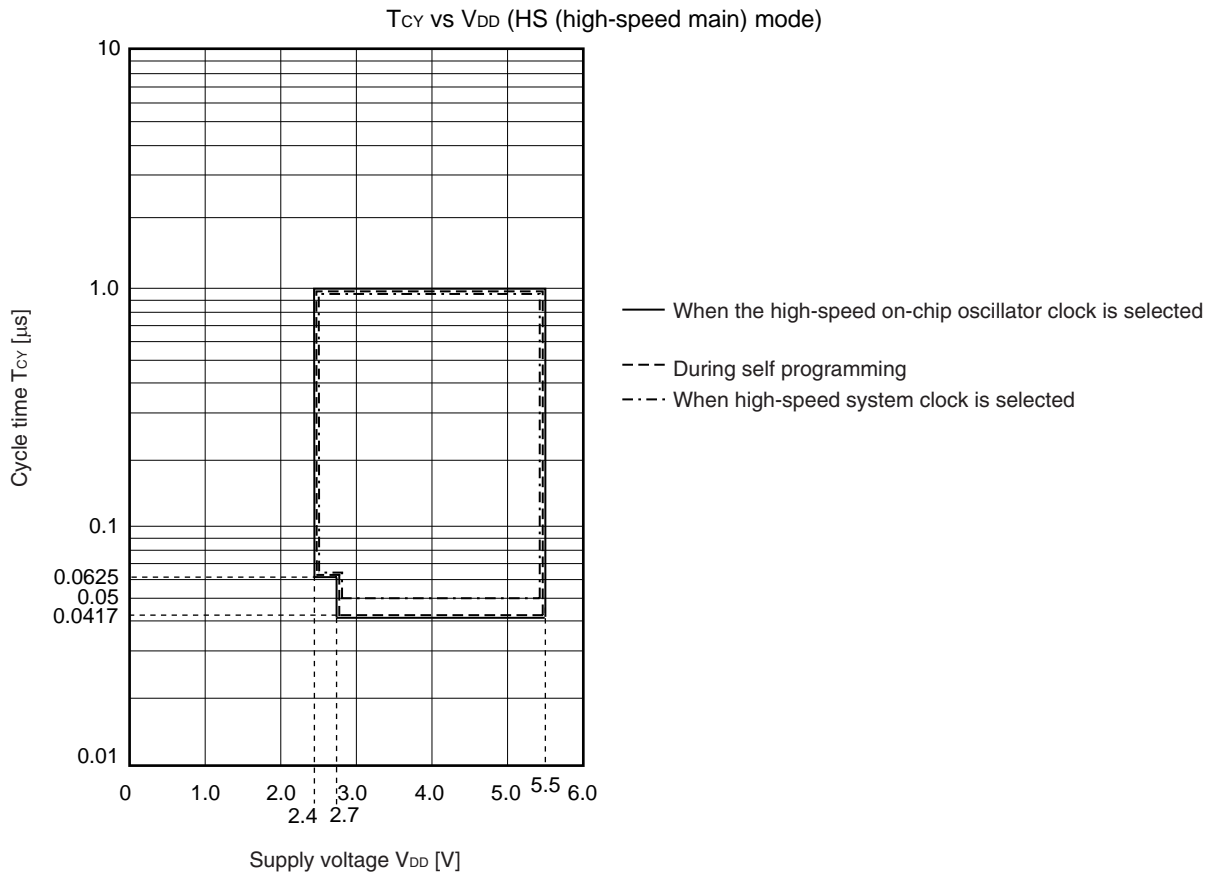
(Note and Remark are listed on the next page.)



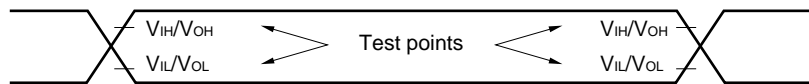
**Note** Specification under conditions where the duty factor is 50%.

**Remark** f<sub>MCK</sub>: Timer array unit operation clock frequency  
 (Operation clock to be set by the CKS<sub>mn0</sub>, CKS<sub>mn1</sub> bits of timer mode register mn (TMR<sub>mn</sub>)  
 m: Unit number (m = 0), n: Channel number (n = 0 to 7))

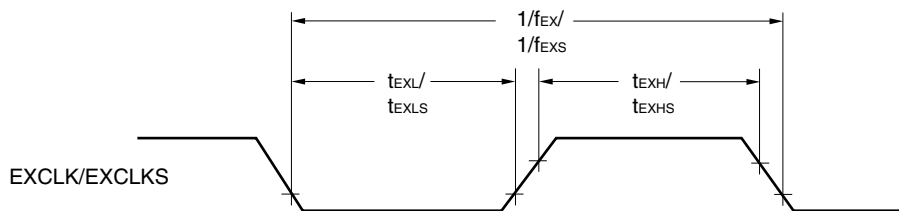
**Minimum Instruction Execution Time during Main System Clock Operation**



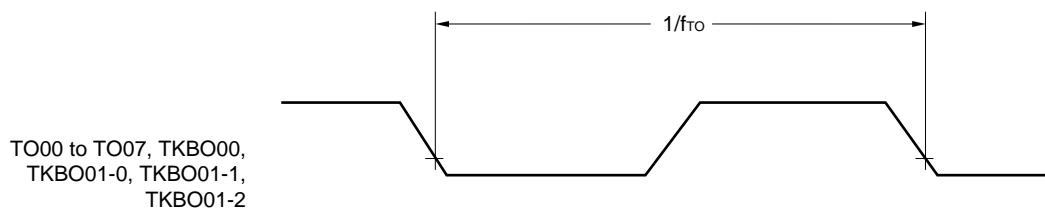
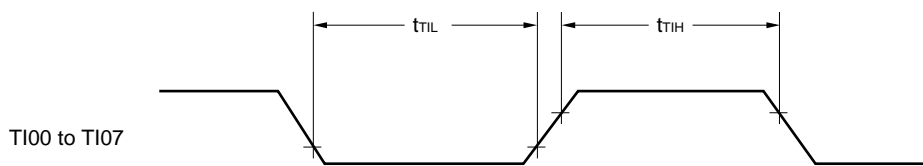
**AC Timing Test Points**



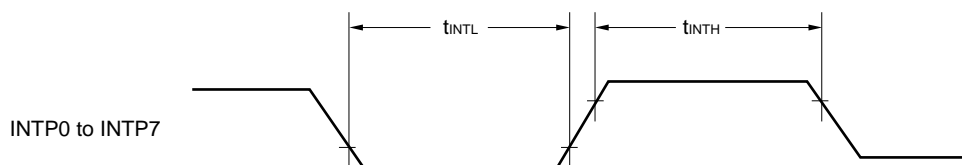
**External System Clock Timing**



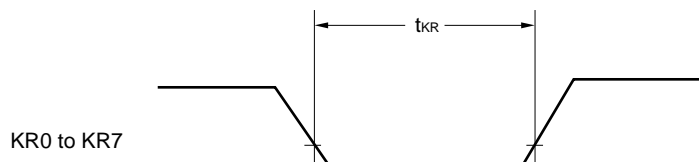
**TI/TO Timing**



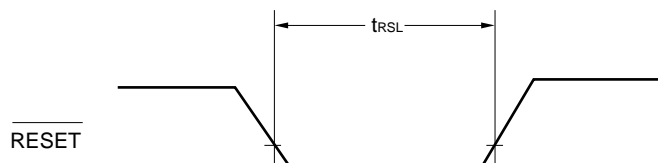
**Interrupt Request Input Timing**



**Key Interrupt Input Timing**

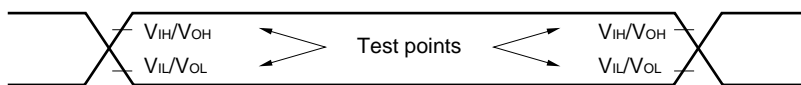


**RESET Input Timing**



### 33.5 Peripheral Functions Characteristics

#### AC Timing Test Points



#### 33.5.1 Serial array unit

##### (1) During communication at same potential (UART mode)

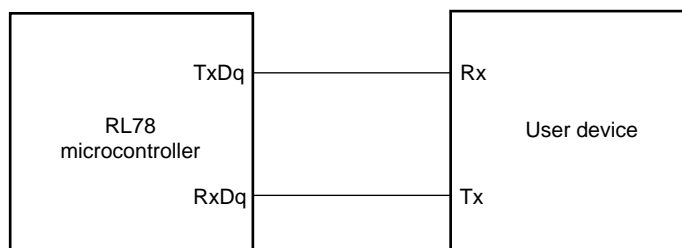
(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
Transfer rate <sup>Note</sup>		Theoretical value of the maximum transfer rate f <sub>CLK</sub> = 24 MHz, f <sub>MCK</sub> = f <sub>CLK</sub>		f <sub>MCK</sub> /12	bps
				2.0	Mbps

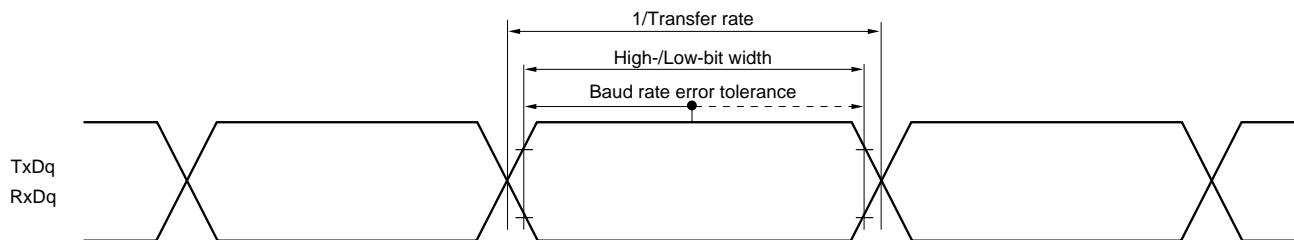
**Note** Transfer rate in the SNOOZE mode is 4800 bps only.

**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 to 3), g: PIM and POM number (g = 0, 1, 3)
  2. f<sub>MCK</sub>: Serial array unit operation clock frequency  
(Operation clock to be set by the CKS<sub>mn</sub> bit of serial mode register mn (SMR<sub>mn</sub>). m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13))

**(2) During communication at same potential (CSI mode) (master mode, SCKp... internal clock output)**  
**( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCKp cycle time	$t_{KCY1}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	334 <sup>Note 1</sup>		ns
		$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	500 <sup>Note 1</sup>		ns
SCKp high-/low-level width	$t_{KH1}$ , $t_{KL1}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$t_{KCY1}/2 - 24$		ns
		$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$t_{KCY1}/2 - 36$		ns
		$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$t_{KCY1}/2 - 76$		ns
Slp setup time (to SCKp $\uparrow$ ) <sup>Note 2</sup>	$t_{SIK1}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	66		ns
		$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	66		ns
		$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	113		ns
Slp hold time (from SCKp $\uparrow$ ) <sup>Note 3</sup>	$t_{SH1}$		38		ns
Delay time from SCKp $\downarrow$ to SOp output <sup>Note 4</sup>	$t_{KSO1}$	$C = 30\text{ pF}$ <sup>Note 5</sup>		50	ns

**Notes 1.** The value must also be equal to or more than  $4/f_{CLK}$ .

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The Slp setup time becomes “to SCKp $\downarrow$ ” 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 Slp hold time becomes “from SCKp $\downarrow$ ” 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 delay time to SOp output becomes “from SCKp $\uparrow$ ” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.
5. C is the load capacitance of the SCKp and SOp output lines.

**Caution** Select the normal input buffer for the Slp 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, 10), m: Unit number (m = 0), n: Channel number (n = 0, 2),  
g: PIM and POM numbers (g = 0, 1)
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, 02))

**(3) During communication at same potential (CSI mode) (slave mode, SCKp... external clock input)**  
**(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)**

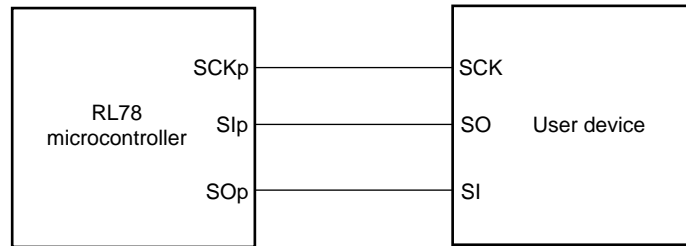
Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time <sup>Note 5</sup>	t <sub>KCY2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V	f <sub>MCK</sub> > 20 MHz	16/f <sub>MCK</sub>		ns
			f <sub>MCK</sub> ≤ 20 MHz	12/f <sub>MCK</sub>		ns
		2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	f <sub>MCK</sub> > 16 MHz	16/f <sub>MCK</sub>		ns
			f <sub>MCK</sub> ≤ 16 MHz	12/f <sub>MCK</sub>		ns
	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V	12/f <sub>MCK</sub> and 1000		ns		
SCKp high-/low-level width	t <sub>KH2</sub> , t <sub>KL2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V		t <sub>KCY2</sub> /2-14		ns
		2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V		t <sub>KCY2</sub> /2-16		ns
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		t <sub>KCY2</sub> /2-36		ns
Slp setup time (to SCKp↑) <sup>Note 1</sup>	t <sub>SIK2</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V		1/f <sub>MCK</sub> +40		ns
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		1/f <sub>MCK</sub> +60		ns
Slp hold time (from SCKp↑) <sup>Note 2</sup>	t <sub>KSI2</sub>			1/f <sub>MCK</sub> +62		ns
Delay time from SCKp↓ to SOp output <sup>Note 3</sup>	t <sub>KSO2</sub>	C = 30 pF <sup>Note 4</sup>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V		2/f <sub>MCK</sub> +66	ns
			2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		2/f <sub>MCK</sub> +113	ns

- Notes**
1. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The Slp setup time becomes “to SCKp↓” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
  2. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The Slp hold time becomes “from SCKp↓” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
  3. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The delay time to SOp output becomes “from SCKp↑” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
  4. C is the load capacitance of the SOp output lines.
  5. Transfer rate in SNOOZE mode: MAX. 1 Mbps

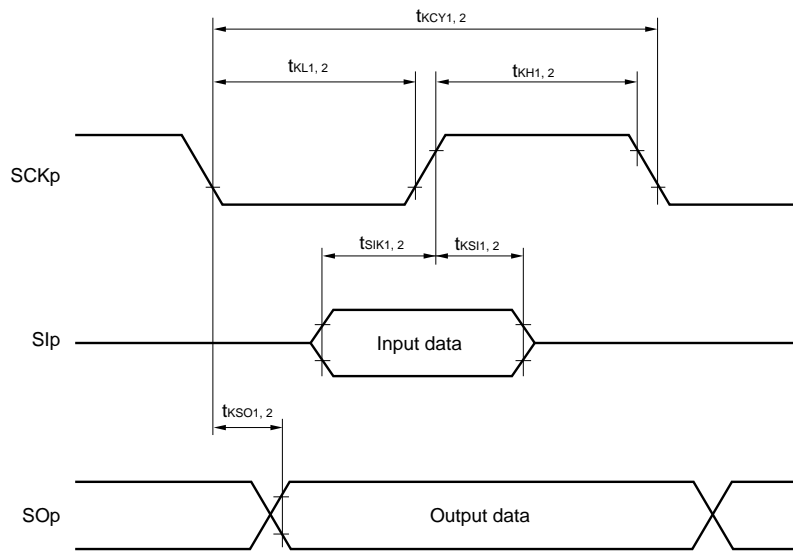
**Caution** Select the normal input buffer for the Slp pin and SCKp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

- Remarks**
1. p: CSI number (p = 00, 10), m: Unit number (m = 0), n: Channel number (n = 0, 2),  
g: PIM number (g = 0, 1)
  2. f<sub>MCK</sub>: Serial array unit operation clock frequency  
(Operation clock to be set by the CKS<sub>mn</sub> bit of serial mode register mn (SMR<sub>mn</sub>). m: Unit number,  
n: Channel number (mn = 00, 02))

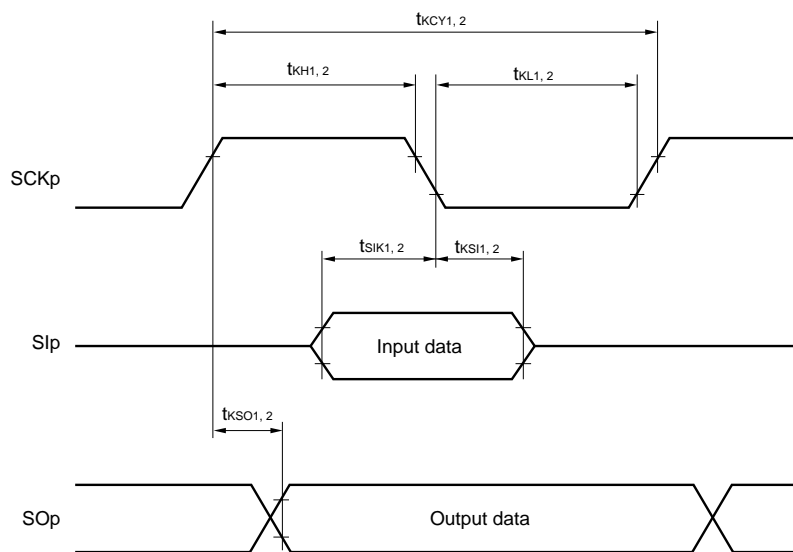
**CSI mode connection diagram (during communication at same potential)**



**CSI mode serial transfer timing (during communication at same potential)  
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)**



**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, 10)
  2. m: Unit number, n: Channel number (mn = 00, 02)

**(4) During communication at same potential (simplified I<sup>2</sup>C mode)****(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)**

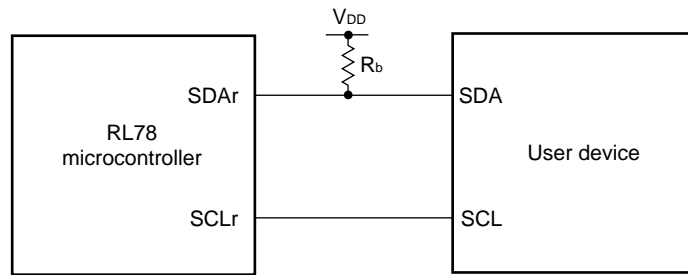
Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f <sub>SCL</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ		400 <sup>Note 1</sup>	kHz
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ		100 <sup>Note 1</sup>	
Hold time when SCLr = "L"	t <sub>LOW</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	1200		ns
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	4600		
Hold time when SCLr = "H"	t <sub>HIGH</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	1200		ns
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	4600		
Data setup time (reception)	t <sub>SU:DAT</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	1/f <sub>MCK</sub> + 220 <sup>Note 2</sup>		ns
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	1/f <sub>MCK</sub> + 580 <sup>Note 2</sup>		
Data hold time (transmission)	t <sub>HD:DAT</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	0	770	ns
		2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	0	1420	

- Notes**
1. The value must also be equal to or less than f<sub>MCK</sub>/4.
  2. Set the f<sub>MCK</sub> value to keep the hold time of SCLr = "L" and SCLr = "H".

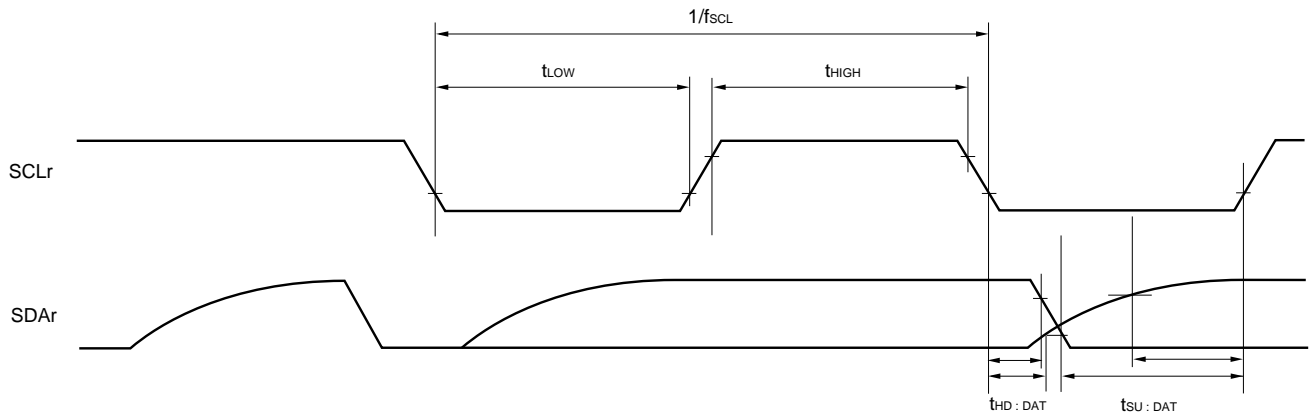
**Caution** Select the normal input buffer and the N-ch open drain output (V<sub>DD</sub> tolerance) mode for the SDAr pin and the normal output mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg).

(Remarks are listed on the next page.)

**Simplified I<sup>2</sup>C mode connection diagram (during communication at same potential)**



**Simplified I<sup>2</sup>C mode serial transfer timing (during communication at same potential)**



**Remarks 1.** R<sub>b</sub>[Ω]: Communication line (SDAr) pull-up resistance, C<sub>b</sub>[F]: Communication line (SDAr, SCLr) load capacitance

**2.** r: IIC number (r = 00, 10), g: PIM and POM number (g = 0, 1)

**3.** f<sub>MCK</sub>: 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 = 0-3), mn = 00-03, 10-13)

<R>



**(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode) (1/2)****(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)**

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit	
			MIN.	MAX.		
Transfer rate		Reception	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V		f <sub>mck</sub> /12 <sup>Note</sup>	bps
			Theoretical value of the maximum transfer rate f <sub>CLK</sub> = 24 MHz, f <sub>mck</sub> = f <sub>CLK</sub>		2.0	Mbps
			2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V		f <sub>mck</sub> /12 <sup>Note</sup>	bps
			Theoretical value of the maximum transfer rate f <sub>CLK</sub> = 24 MHz, f <sub>mck</sub> = f <sub>CLK</sub>		2.0	Mbps
			2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V		f <sub>mck</sub> /12 <sup>Note</sup>	bps
			Theoretical value of the maximum transfer rate f <sub>CLK</sub> = 24 MHz, f <sub>mck</sub> = f <sub>CLK</sub>		2.0	Mbps

**Note** Transfer rate in SNOOZE mode is 4800 bps only.

**Caution** Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V<sub>DD</sub> tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V<sub>IH</sub> and V<sub>IL</sub>, see the DC characteristics with TTL input buffer selected.

- Remarks**
1. V<sub>b</sub>[V]: Communication line voltage
  2. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0, 1, 3)
  3. f<sub>mck</sub>: 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 to 03, 10 to 13))

**(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode) (2/2)**(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit	
			MIN.	MAX.		
Transfer rate		Transmission	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V		<b>Note 1</b>	bps
			Theoretical value of the maximum transfer rate C <sub>b</sub> = 50 pF, R <sub>b</sub> = 1.4 kΩ, V <sub>b</sub> = 2.7 V		2.0 <sup>Note 2</sup>	Mbps
			2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V		<b>Note 3</b>	bps
			Theoretical value of the maximum transfer rate C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ, V <sub>b</sub> = 2.3 V		1.2 <sup>Note 4</sup>	Mbps
			2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V		<b>Note 5</b>	bps
			Theoretical value of the maximum transfer rate C <sub>b</sub> = 50 pF, R <sub>b</sub> = 5.5 kΩ, V <sub>b</sub> = 1.6 V		0.43 <sup>Note 6</sup>	Mbps

**Notes 1.** The smaller maximum transfer rate derived by using f<sub>MCK</sub>/12 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V ≤ V<sub>DD</sub> ≤ 5.5 V and 2.7 V ≤ V<sub>b</sub> ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

\* 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<sub>MCK</sub>/12 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ V<sub>DD</sub> < 4.0 V and 2.3 V ≤ V<sub>b</sub> ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

\* 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.

**Notes 5.** 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  $2.4\text{ V} \leq V_{DD} < 3.3\text{ V}$  and  $1.6\text{ V} \leq V_b \leq 2.0\text{ V}$

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

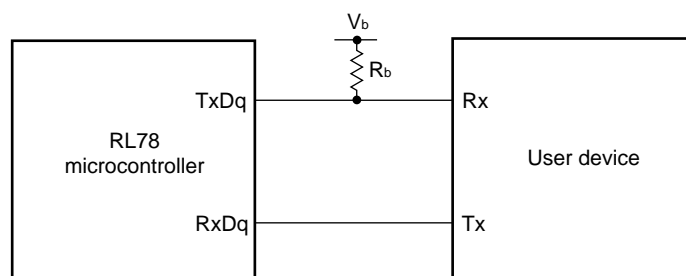
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

\* This value is the theoretical value of the relative difference between the transmission and reception sides.

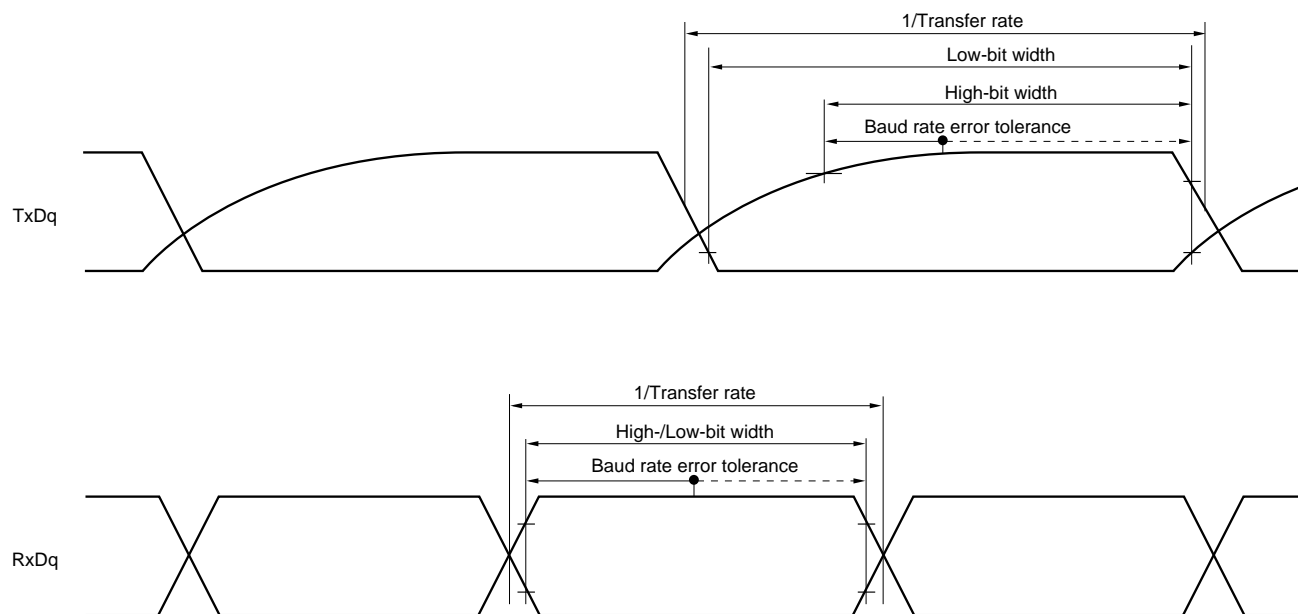
6. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 5** above to calculate the maximum transfer rate under conditions of the customer.

**Caution** Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V<sub>DD</sub> tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V<sub>IH</sub> and V<sub>IL</sub>, 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.  $R_b[\Omega]$ : Communication line (TxDq) pull-up resistance,  $C_b[\text{F}]$ : Communication line (TxDq) load capacitance,  $V_b[\text{V}]$ : Communication line voltage
  2. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0, 1, 3)
  3.  $f_{\text{MCK}}$ : 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 to 03, 10 to 13))

**(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output) (1/2)**  
**(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)**

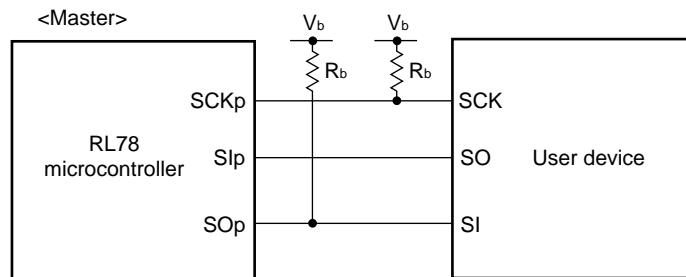
Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCKp cycle time	t <sub>KCY1</sub>	t <sub>KCY1</sub> ≥ 4/f <sub>CLK</sub> 4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	600		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	1000		ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 1.8 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	2300		ns
SCKp high-level width	t <sub>KH1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	t <sub>KCY1</sub> /2 - 150		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	t <sub>KCY1</sub> /2 - 340		ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	t <sub>KCY1</sub> /2 - 916		ns
SCKp low-level width	t <sub>KL1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	t <sub>KCY1</sub> /2 - 24		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	t <sub>KCY1</sub> /2 - 36		ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	t <sub>KCY1</sub> /2 - 100		ns
Slp setup time (to SCKp↑) <sup>Note 1</sup>	t <sub>SIK1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	162		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	354		ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	958		ns
Slp hold time (from SCKp↑) <sup>Note 1</sup>	t <sub>KSH1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	38		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	38		ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	38		ns
Delay time from SCKp↓ to SOp output <sup>Note 1</sup>	t <sub>KSO1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ		200	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ		390	ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ		966	ns

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

**(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output) (2/2)**  
 (T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
Slp setup time (to SCKp↓) <sup>Note 2</sup>	t <sub>SIK1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 1.4 kΩ	88		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 2.7 kΩ	88		ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	220		ns
Slp hold time (from SCKp↓) <sup>Note 2</sup>	t <sub>KSI1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 1.4 kΩ	38		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 2.7 kΩ	38		ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ	38		ns
Delay time from SCKp↑ to SO <sub>p</sub> output <sup>Note 2</sup>	t <sub>KSO1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 1.4 kΩ		50	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 20 pF, R <sub>b</sub> = 2.7 kΩ		50	ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ		50	ns

**CSI mode connection diagram (during communication at different potential)**

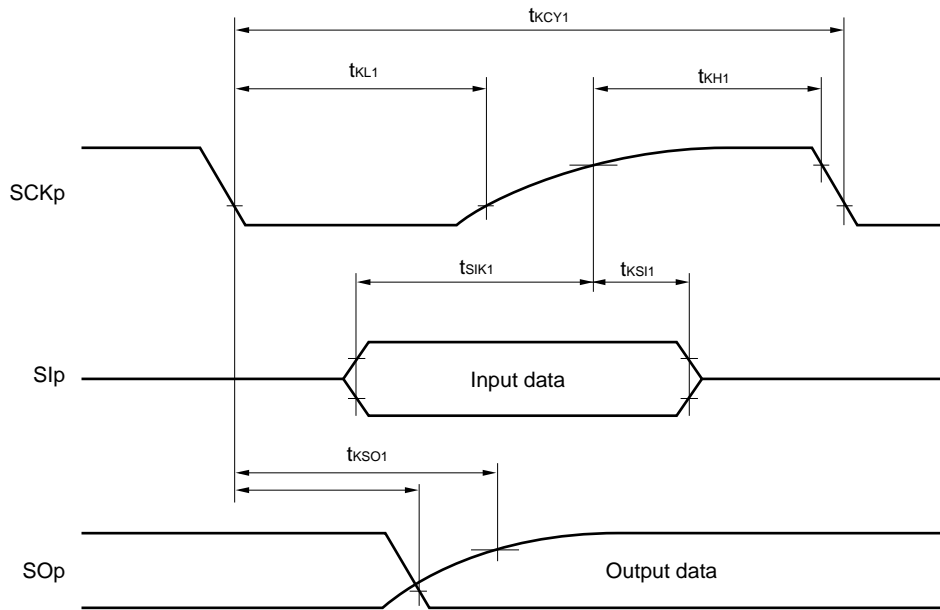


- Notes**
1. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1.
  2. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.

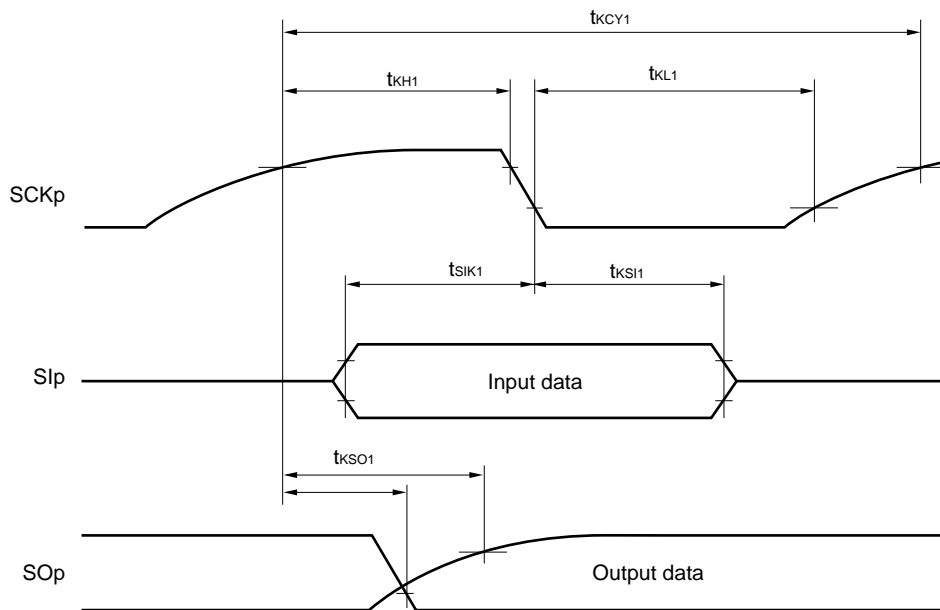
**Caution** Select the TTL input buffer for the Slp pin and the N-ch open drain output (V<sub>DD</sub> tolerance) mode for the SO<sub>p</sub> pin and SCK<sub>p</sub> pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V<sub>IH</sub> and V<sub>IL</sub>, see the DC characteristics with TTL input buffer selected.

- Remarks**
1. R<sub>b</sub>[Ω]: Communication line (SCK<sub>p</sub>, SO<sub>p</sub>) pull-up resistance, C<sub>b</sub>[F]: Communication line (SCK<sub>p</sub>, SO<sub>p</sub>) load capacitance, V<sub>b</sub>[V]: Communication line voltage
  2. p: CSI number (p = 00, 10), m: Unit number, n: Channel number (mn = 00, 02),  
g: PIM and POM number (g = 0, 1)
  3. f<sub>MCK</sub>: Serial array unit operation clock frequency  
(Operation clock to be set by the CKS<sub>mn</sub> bit of serial mode register mn (SMR<sub>mn</sub>).  
m: Unit number, n: Channel number (mn = 00))

**CSI mode serial transfer timing (master mode) (during communication at different potential)  
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)**



**CSI mode serial transfer timing (master mode) (during communication at different potential)  
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)**



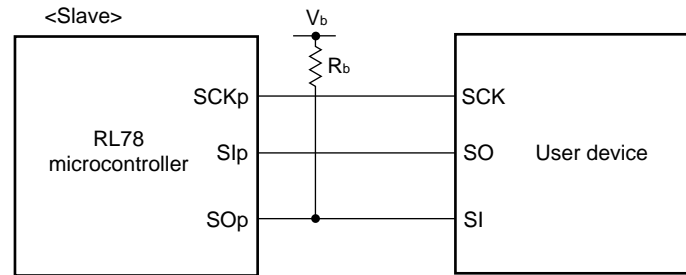
**Remark** p: CSI number (p = 00, 10), m: Unit number, n: Channel number (mn = 00, 02),  
g: PIM and POM number (g = 0, 1)

**(7) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (slave mode, SCKp... external clock input)**  
**(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)**

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCKp cycle time <sup>Note 1</sup>	t <sub>KCY2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V	20 MHz < f <sub>MCK</sub>	24/f <sub>MCK</sub>	ns
			8 MHz < f <sub>MCK</sub> ≤ 20 MHz	20/f <sub>MCK</sub>	ns
			4 MHz < f <sub>MCK</sub> ≤ 8 MHz	16/f <sub>MCK</sub>	ns
			f <sub>MCK</sub> ≤ 4 MHz	12/f <sub>MCK</sub>	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V	20 MHz < f <sub>MCK</sub>	32/f <sub>MCK</sub>	ns
			16 MHz < f <sub>MCK</sub> ≤ 20 MHz	28/f <sub>MCK</sub>	ns
			8 MHz < f <sub>MCK</sub> ≤ 16 MHz	24/f <sub>MCK</sub>	ns
			4 MHz < f <sub>MCK</sub> ≤ 8 MHz	16/f <sub>MCK</sub>	ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V	f <sub>MCK</sub> ≤ 4 MHz	12/f <sub>MCK</sub>	ns
			20 MHz < f <sub>MCK</sub>	72/f <sub>MCK</sub>	ns
			16 MHz < f <sub>MCK</sub> ≤ 20 MHz	64/f <sub>MCK</sub>	ns
			8 MHz < f <sub>MCK</sub> ≤ 16 MHz	52/f <sub>MCK</sub>	ns
SCKp high-/low-level width	t <sub>KH2</sub> , t <sub>KL2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V	t <sub>KCY2</sub> /2 - 24	ns	
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V	t <sub>KCY2</sub> /2 - 36	ns	
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V	t <sub>KCY2</sub> /2 - 100	ns	
Slp setup time (to SCKp↑) <sup>Note 2</sup>	t <sub>SIK2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V	1/f <sub>MCK</sub> + 40	ns	
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V	1/f <sub>MCK</sub> + 40	ns	
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V	1/f <sub>MCK</sub> + 60	ns	
Slp hold time (from SCKp↑) <sup>Note 3</sup>	t <sub>KSJ2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V	1/f <sub>MCK</sub> + 62	ns	
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V	1/f <sub>MCK</sub> + 62	ns	
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V	1/f <sub>MCK</sub> + 62	ns	
Delay time from SCKp↓ to SOp output <sup>Note 4</sup>	t <sub>KSO2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ		2/f <sub>MCK</sub> + 240	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ		2/f <sub>MCK</sub> + 428	ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 5.5 kΩ		2/f <sub>MCK</sub> + 1146	ns

(Notes and Caution are listed on the next page, and Remarks are listed on the page after the next page.)

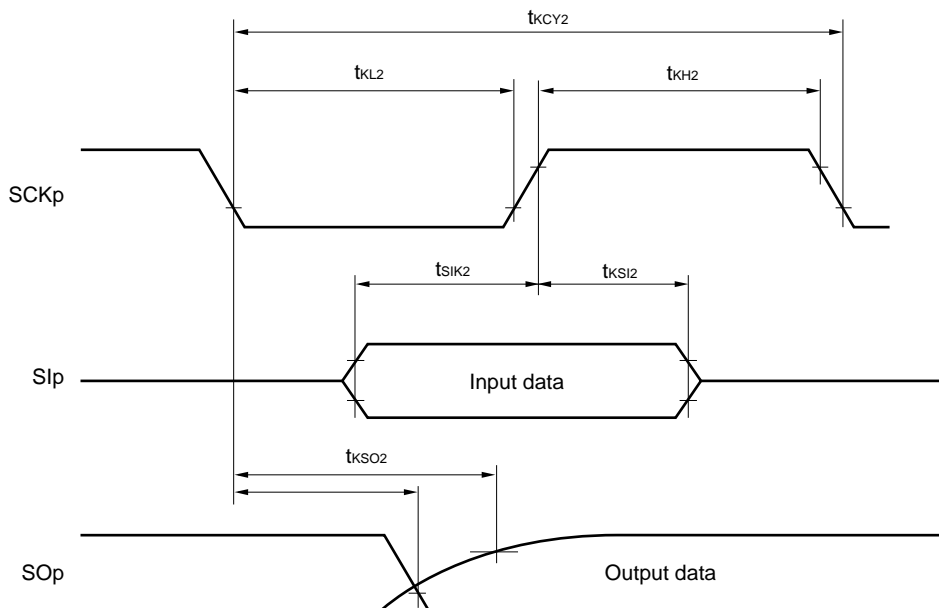


**CSI mode connection diagram (during communication at different potential)**

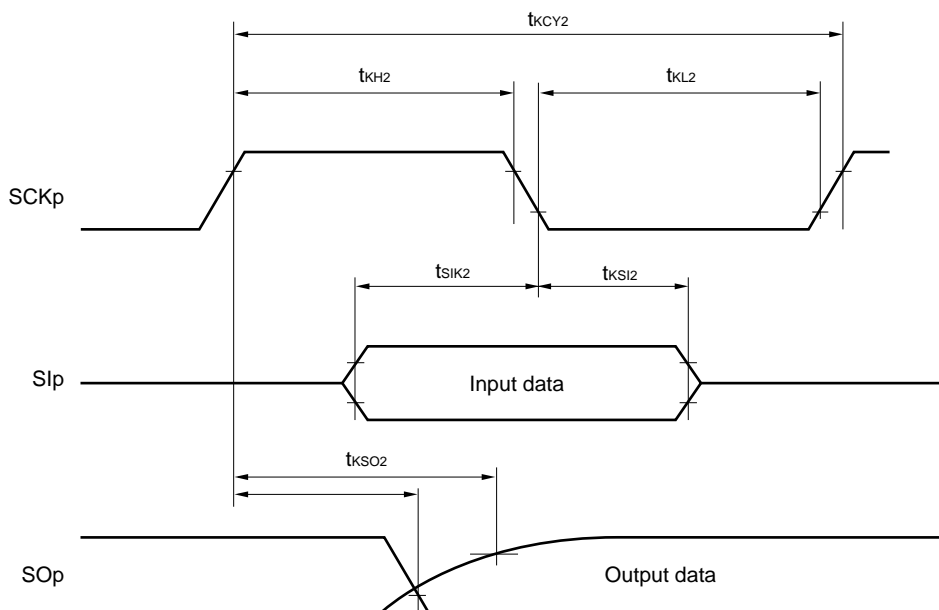
- Notes**
1. Transfer rate in SNOOZE mode: MAX. 1 Mbps
  2. When  $DAPmn = 0$  and  $CKPmn = 0$ , or  $DAPmn = 1$  and  $CKPmn = 1$ . The  $Slp$  setup time becomes “to  $SCKp\downarrow$ ” when  $DAPmn = 0$  and  $CKPmn = 1$ , or  $DAPmn = 1$  and  $CKPmn = 0$ .
  3. When  $DAPmn = 0$  and  $CKPmn = 0$ , or  $DAPmn = 1$  and  $CKPmn = 1$ . The  $Slp$  hold time becomes “from  $SCKp\downarrow$ ” 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 delay time to  $SOp$  output becomes “from  $SCKp\uparrow$ ” when  $DAPmn = 0$  and  $CKPmn = 1$ , or  $DAPmn = 1$  and  $CKPmn = 0$ .

**Caution** Select the TTL input buffer for the  $Slp$  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.

**CSI mode serial transfer timing (slave mode) (during communication at different potential)  
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)**



**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.**  $R_b[\Omega]$ : Communication line (SO<sub>p</sub>) pull-up resistance,  $C_b[\text{F}]$ : Communication line (SO<sub>p</sub>) load capacitance,  $V_b[\text{V}]$ : Communication line voltage
- 2.** p: CSI number (p = 00, 10), m: Unit number, n: Channel number (mn = 00, 02), g: PIM and POM number (g = 0, 1)
- 3.**  $f_{\text{mck}}$ : Serial array unit operation clock frequency  
(Operation clock to be set by the CKS<sub>m</sub>n bit of serial mode register mn (SMR<sub>m</sub>n))  
m: Unit number, n: Channel number (mn = 00, 02))

**(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I<sup>2</sup>C mode) (1/2)****(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)**

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f <sub>SCL</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ		400 <sup>Note 1</sup>	kHz
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ		400 <sup>Note 1</sup>	kHz
		4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.8 kΩ		100 <sup>Note 1</sup>	kHz
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ		100 <sup>Note 1</sup>	kHz
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5.5 kΩ		100 <sup>Note 1</sup>	kHz
Hold time when SCLr = "L"	t <sub>LOW</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	1200		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	1200		ns
		4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.8 kΩ	4600		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	4600		ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5.5 kΩ	4650		ns
Hold time when SCLr = "H"	t <sub>HIGH</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	620		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> < 2.7 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	500		ns
		4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.8 kΩ	2700		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	2400		ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5.5 kΩ	1830		ns

(Notes and Caution are listed on the next page, and Remarks are listed on the page after the next page.)

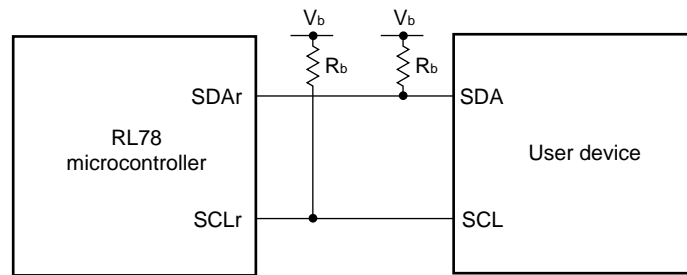
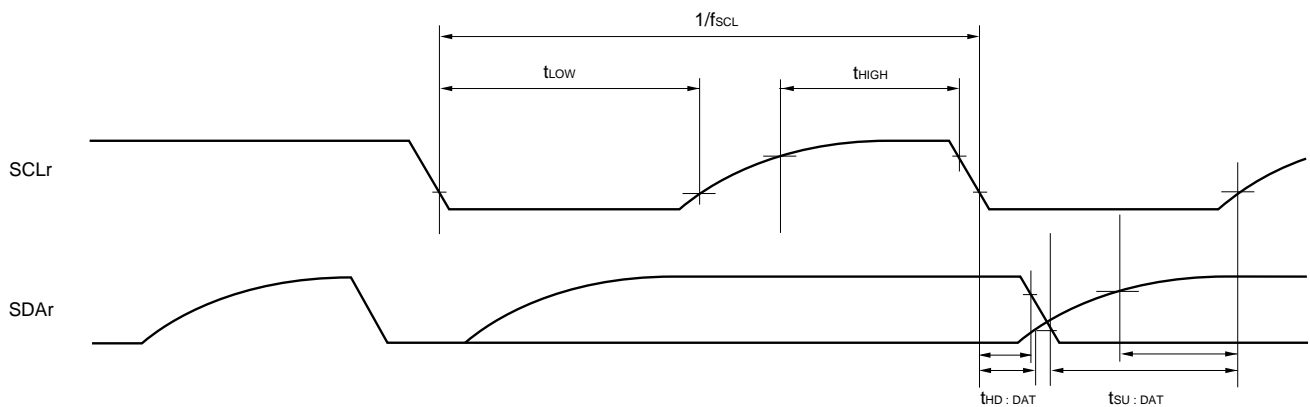
**(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I<sup>2</sup>C mode) (2/2)****(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)**

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
Data setup time (reception)	t <sub>SU:DAT</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	1/f <sub>MCK</sub> + 340 <sup>Note 2</sup>		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	1/f <sub>MCK</sub> + 340 <sup>Note 2</sup>		ns
		4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.8 kΩ	1/f <sub>MCK</sub> + 760 <sup>Note 2</sup>		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	1/f <sub>MCK</sub> + 760 <sup>Note 2</sup>		ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5.5 kΩ	1/f <sub>MCK</sub> + 570 <sup>Note 2</sup>		ns
Data hold time (transmission)	t <sub>HD:DAT</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	0	770	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 kΩ	0	770	ns
		4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.8 kΩ	0	1420	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	0	1420	ns
		2.4 V ≤ V <sub>DD</sub> < 3.3 V, 1.6 V ≤ V <sub>b</sub> ≤ 2.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5.5 kΩ	0	1215	ns

**Notes 1.** The value must also be equal to or less than f<sub>MCK</sub>/4.**2.** Set the f<sub>MCK</sub> 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<sub>DD</sub> tolerance) mode for the SDAr pin and the N-ch open drain output (V<sub>DD</sub> tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V<sub>IH</sub> and V<sub>IL</sub>, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

Simplified I<sup>2</sup>C mode connection diagram (during communication at different potential)Simplified I<sup>2</sup>C mode serial transfer timing (during communication at different potential)

- Remarks 1.**  $R_b[\Omega]$ : 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, 10), g: PIM, POM number (g = 0, 1)
- 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, 02))

33.5.2 Serial interface IICA

(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode				Unit
			Standard Mode		Fast Mode		
			MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f <sub>SCL</sub>	Fast mode: f <sub>CLK</sub> ≥ 3.5 MHz	-	-	0	400	kHz
		Normal mode: f <sub>CLK</sub> ≥ 1 MHz	0	100	-	-	
Setup time of restart condition	t <sub>SU:STA</sub>		4.7		0.6		μs
Hold time <sup>Note 1</sup>	t <sub>HD:STA</sub>		4.0		0.6		μs
Hold time when SCLA0 = "L"	t <sub>LOW</sub>		4.7		1.3		μs
Hold time when SCLA0 = "H"	t <sub>HIGH</sub>		4.0		0.6		μs
Data setup time (reception)	t <sub>SU:DAT</sub>		250		100		ns
Data hold time (transmission) <sup>Note 2</sup>	t <sub>HD:DAT</sub>		0 <sup>Note 3</sup>	3.45	0 <sup>Note 3</sup>	0.9	μs
Setup time of stop condition	t <sub>SU:STO</sub>		4.0		0.6		μs
Bus-free time	t <sub>BUF</sub>		4.7		1.3		μs

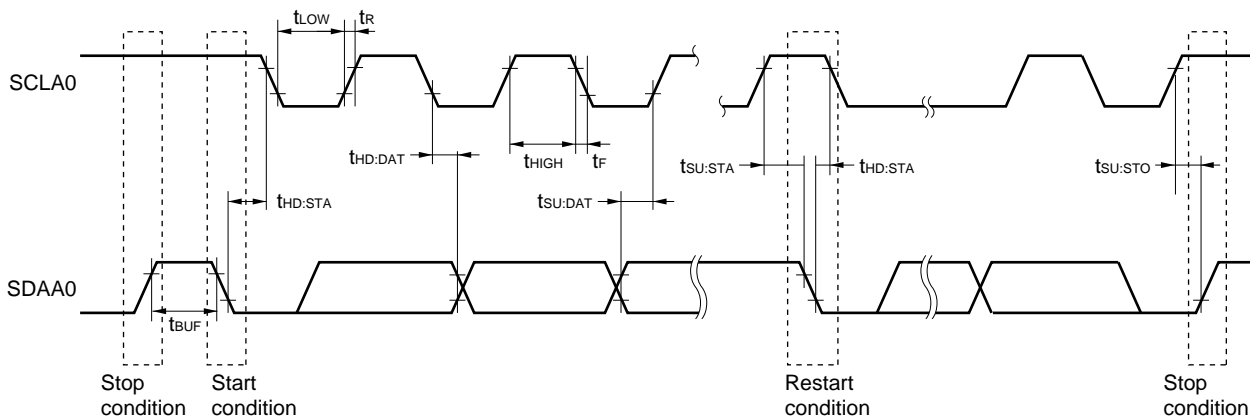
- Notes**
1. The first clock pulse is generated after this period when the start/restart condition is detected.
  2. The maximum value (MAX.) of t<sub>HD:DAT</sub> is during normal transfer and a wait state is inserted in the ACK (acknowledge) timing.

**Remark** The maximum value of C<sub>b</sub> (communication line capacitance) and the value of R<sub>b</sub> (communication line pull-up resistor) at that time in each mode are as follows.

Standard mode: C<sub>b</sub> = 400 pF, R<sub>b</sub> = 2.7 kΩ

Fast mode: C<sub>b</sub> = 320 pF, R<sub>b</sub> = 1.1 kΩ

IICA serial transfer timing



### 33.6 Analog Characteristics

#### 33.6.1 A/D converter characteristics

##### Classification of A/D converter characteristics

Reference Voltage Input channel	Reference voltage (+) = AV <sub>REFP</sub> Reference voltage (-) = AV <sub>REFM</sub>	Reference voltage (+) = V <sub>DD</sub> Reference voltage (-) = V <sub>SS</sub>	Reference voltage (+) = V <sub>BGR</sub> Reference voltage (-) = AV <sub>REFM</sub>
ANI0, ANI1	–	See 33.6.1 (2).	See 33.6.1 (3).
ANI16 to ANI25	See 33.6.1 (1).		
Internal reference voltage Temperature sensor output voltage	See 33.6.1 (1).		

(1) When reference voltage (+) = AV<sub>REFP</sub>/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV<sub>REFM</sub>/ANI1 (ADREFM = 1), target pins: ANI16 to ANI25, internal reference voltage, and temperature sensor output voltage

(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V, Reference voltage (+) = AV<sub>REFP</sub>, Reference voltage (-) = AV<sub>REFM</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES		8		10	bit
Overall error <sup>Note 1</sup>	AINL	10-bit resolution AV <sub>REFP</sub> = V <sub>DD</sub> <sup>Note 3</sup>	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V	1.2	±5.0	LSB
Conversion time	t <sub>CONV</sub>	10-bit resolution Target pin: ANI16 to ANI25	3.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	2.125	39	μs
			2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	3.1875	39	μs
			2.4 V ≤ V <sub>DD</sub> ≤ 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 <sub>DD</sub> ≤ 5.5 V	2.375	39	μs
			2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	3.5625	39	μs
			2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V	17	39	μs
Zero-scale error <sup>Notes 1, 2</sup>	E <sub>ZS</sub>	10-bit resolution AV <sub>REFP</sub> = V <sub>DD</sub> <sup>Note 3</sup>	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		±0.35	%FSR
Full-scale error <sup>Notes 1, 2</sup>	E <sub>FS</sub>	10-bit resolution AV <sub>REFP</sub> = V <sub>DD</sub> <sup>Note 3</sup>	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		±0.35	%FSR
Integral linearity error <sup>Note 1</sup>	ILE	10-bit resolution AV <sub>REFP</sub> = V <sub>DD</sub> <sup>Note 3</sup>	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		±3.5	LSB
Differential linearity error <sup>Note 1</sup>	DLE	10-bit resolution AV <sub>REFP</sub> = V <sub>DD</sub> <sup>Note 3</sup>	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		±2.0	LSB
Analog input voltage	V <sub>AIN</sub>	ANI16 to ANI25	0		AV <sub>REFP</sub>	V
		Internal reference voltage (2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V, HS (high-speed main) mode))			V <sub>BGR</sub> <sup>Note 4</sup>	V
		Temperature sensor output voltage (2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V, HS (high-speed main) mode))			V <sub>TMPS25</sub> <sup>Note 4</sup>	V

(Notes are listed on the next page.)

- Notes**
1. Excludes quantization error ( $\pm 1/2$  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 $\pm 4$ LSB to the MAX. value when $AV_{REFP} = V_{DD}$ .
Zero-scale error/Full-scale error:	Add $\pm 0.2\%$ FSR to the MAX. value when $AV_{REFP} = V_{DD}$ .
Integral linearity error/ Differential linearity error:	Add $\pm 2$ LSB to the MAX. value when $AV_{REFP} = V_{DD}$ .
  4. See **33.6.2 Temperature sensor/internal reference voltage characteristics**.



(2) When reference voltage (+) = V<sub>DD</sub> (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = V<sub>SS</sub> (ADREFM = 0), target pins: ANI0, ANI1, ANI16 to ANI25, internal reference voltage, and temperature sensor output voltage

(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V, Reference voltage (+) = V<sub>DD</sub>, Reference voltage (-) = V<sub>SS</sub>)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error <sup>Note 1</sup>	AINL	10-bit resolution	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V		1.2	±7.0	LSB
Conversion time	t <sub>CONV</sub>	10-bit resolution Target pin: ANI0, ANI1, ANI16 to ANI25	3.6 V ≤ V <sub>DD</sub> ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	3.1875		39	μs
			2.4 V ≤ V <sub>DD</sub> ≤ 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 <sub>DD</sub> ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V	17		39	μs
Zero-scale error <sup>Notes 1, 2</sup>	E <sub>ZS</sub>	10-bit resolution	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V			±0.60	%FSR
Full-scale error <sup>Notes 1, 2</sup>	E <sub>FS</sub>	10-bit resolution	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V			±0.60	%FSR
Integral linearity error <sup>Note 1</sup>	ILE	10-bit resolution	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V			±4.0	LSB
Differential linearity error <sup>Note 1</sup>	DLE	10-bit resolution	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V			±2.0	LSB
Analog input voltage	V <sub>AIN</sub>	ANI0, ANI1, ANI16 to ANI25		0		V <sub>DD</sub>	V
		Internal reference voltage (2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V, HS (high-speed main) mode))		V <sub>BGR</sub> <sup>Note 3</sup>			V
		Temperature sensor output voltage (2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V, HS (high-speed main) mode))		V <sub>TMPS25</sub> <sup>Note 3</sup>			V

**Notes** 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. See 33.6.2 Temperature sensor/internal reference voltage characteristics.

(3) When reference voltage (+) = internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), reference voltage (-) = AVREFM/ANI1 (ADREFM = 1), target pins: ANI0, ANI16 to ANI25

( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ , Reference voltage (+) =  $V_{BGR}$ <sup>Note 3</sup>,  
Reference voltage (-) =  $AV_{REFM}$ <sup>Note 4</sup> = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8			bit
Conversion time	$t_{CONV}$	8-bit resolution	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	17		39	$\mu\text{s}$
Zero-scale error <sup>Notes 1, 2</sup>	E <sub>ZS</sub>	8-bit resolution	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			$\pm 0.60$	%FSR
Integral linearity error <sup>Note 1</sup>	ILE	8-bit resolution	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			$\pm 2.0$	LSB
Differential linearity error <sup>Note 1</sup>	DLE	8-bit resolution	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			$\pm 1.0$	LSB
Analog input voltage	$V_{AIN}$			0		$V_{BGR}$ <sup>Note 3</sup>	V

**Notes** 1. Excludes quantization error ( $\pm 1/2$  LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. See 33.6.2 Temperature sensor/internal reference voltage characteristics.

4. When reference voltage (-) =  $V_{SS}$ , the MAX. values are as follows.

Zero-scale error: Add  $\pm 0.35\%$ FSR to the  $AV_{REFM}$  MAX. value.

Integral linearity error: Add  $\pm 0.5$  LSB to the  $AV_{REFM}$  MAX. value.

Differential linearity error: Add  $\pm 0.2$  LSB to the  $AV_{REFM}$  MAX. value.

### 33.6.2 Temperature sensor/internal reference voltage characteristics

( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ , HS (high-speed main) mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	$V_{TMS25}$	ADS register = 80H, $T_A = +25^\circ\text{C}$		1.05		V
Internal reference output voltage	$V_{BGR}$	ADS register = 81H	1.38	1.45	1.5	V
Temperature coefficient	$F_{VTMS}$	Temperature sensor that depends on the temperature		-3.6		mV/ $^\circ\text{C}$
Operation stabilization wait time	$t_{AMP}$				5	$\mu\text{s}$

### 33.6.3 Comparator

( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input voltage range	$I_{vref}$		0		$V_{DD} - 1.4$	V	
	$I_{vcmp}$		-0.3		$V_{DD} + 0.3$	V	
Output delay	$t_d$	$V_{DD} = 3.0\text{ V}$ Input slew rate $> 50\text{ mV}/\mu\text{s}$	Comparator high-speed mode, standard mode			1.2	$\mu\text{s}$
			Comparator high-speed mode, window mode			2.0	$\mu\text{s}$
			Comparator low-speed mode, standard mode		3.0	5.0	$\mu\text{s}$
High-electric-potential reference voltage	$V_{TW+}$	Comparator high-speed mode, window mode	$0.66V_{DD}$	$0.76V_{DD}$	$0.86V_{DD}$	V	
Low-electric-potential reference voltage	$V_{TW-}$	Comparator high-speed mode, window mode	$0.14V_{DD}$	$0.24V_{DD}$	$0.34V_{DD}$	V	
Operation stabilization wait time	$t_{CMP}$		100			$\mu\text{s}$	
Internal reference output voltage <sup>Note</sup>	$V_{BGR}$	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , HS (high-speed main) mode	1.38	1.45	1.50	V	

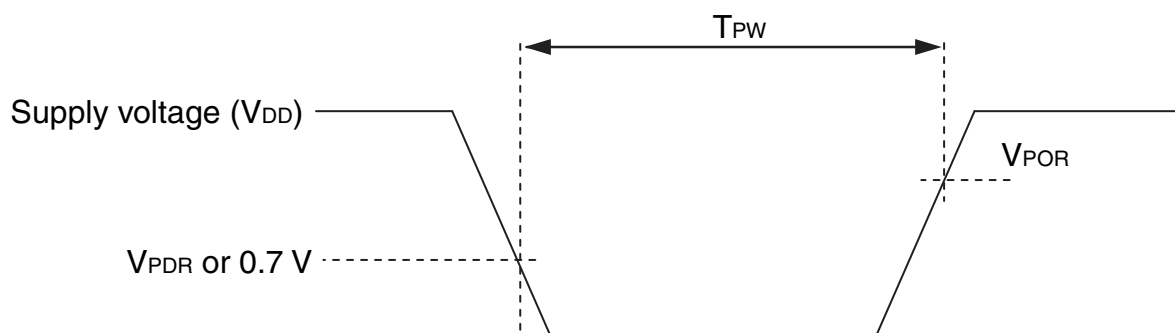
**Note** Cannot be used in subsystem clock operation and STOP mode.

## 33.6.4 POR circuit characteristics

 $(T_A = -40$  to  $+105^\circ\text{C}$ ,  $V_{SS} = 0$  V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	$V_{POR}$	When power supply rises	1.45	1.51	1.57	V
	$V_{PDR}$	When power supply falls	1.44	1.50	1.56	V
Minimum pulse width <sup>Note</sup>	$T_{PW}$		300			$\mu\text{s}$

**Note** This is the time required for the POR circuit to execute a reset operation when  $V_{DD}$  falls below  $V_{PDR}$ . When the microcontroller enters STOP mode and when the main system clock ( $f_{MAIN}$ ) has been stopped by setting bit 0 (HIOSTOP) and bit 7 (MSTOP) of the clock operation status control register (CSC), this is the time required for the POR circuit to execute a reset operation between when  $V_{DD}$  falls below 0.7 V and when  $V_{DD}$  rises to  $V_{POR}$  or higher.



## 33.6.5 LVD circuit characteristics

## LVD Detection Voltage of Reset Mode and Interrupt Mode

( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $V_{PDR} \leq V_{DD} \leq 5.5$  V,  $V_{SS} = 0$  V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V <sub>LVD0</sub>	When power supply rises	3.90	4.06	4.22	V
			When power supply falls	3.83	3.98	4.13	V
		V <sub>LVD1</sub>	When power supply rises	3.60	3.75	3.90	V
			When power supply falls	3.53	3.67	3.81	V
		V <sub>LVD2</sub>	When power supply rises	3.01	3.13	3.25	V
			When power supply falls	2.94	3.06	3.18	V
		V <sub>LVD3</sub>	When power supply rises	2.90	3.02	3.14	V
			When power supply falls	2.85	2.96	3.07	V
		V <sub>LVD4</sub>	When power supply rises	2.81	2.92	3.03	V
			When power supply falls	2.75	2.86	2.97	V
		V <sub>LVD5</sub>	When power supply rises	2.71	2.81	2.92	V
			When power supply falls	2.64	2.75	2.86	V
		V <sub>LVD6</sub>	When power supply rises	2.61	2.71	2.81	V
			When power supply falls	2.55	2.65	2.75	V
		V <sub>LVD7</sub>	When power supply rises	2.51	2.61	2.71	V
			When power supply falls	2.45	2.55	2.65	V
Minimum pulse width		t <sub>LW</sub>		300			μs
Detection delay time						300	μs

**LVD Detection Voltage of Interrupt & Reset Mode****( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $V_{PDR} \leq V_{DD} \leq 5.5$  V,  $V_{SS} = 0$  V)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Interrupt and reset mode	$V_{LVD5}$	$V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 1$ , falling reset voltage	2.64	2.75	2.86	V	
	$V_{LVD4}$	$LVIS1, LVIS0 = 1, 0$	Rising release reset voltage	2.81	2.92	3.03	V
			Falling interrupt voltage	2.75	2.86	2.97	V
	$V_{LVD3}$	$LVIS1, LVIS0 = 0, 1$	Rising release reset voltage	2.90	3.02	3.14	V
			Falling interrupt voltage	2.85	2.96	3.07	V
	$V_{LVD0}$	$LVIS1, LVIS0 = 0, 0$	Rising release reset voltage	3.90	4.06	4.22	V
Falling interrupt voltage			3.83	3.98	4.13	V	

**33.6.6 Supply voltage rise time****( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $V_{SS} = 0$  V)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$V_{DD}$ rise slope	$SV_{DD}$				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 33.4 AC Characteristics.

### 33.7 LCD Characteristics

#### 33.7.1 External resistance division method

##### (1) Static display mode

( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $V_{L4} (\text{MIN.}) \leq V_{DD} \leq 5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	$V_{L4}$		2.0		$V_{DD}$	V

##### (2) 1/2 bias method, 1/4 bias method

( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $V_{L4} (\text{MIN.}) \leq V_{DD} \leq 5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	$V_{L4}$		2.7		$V_{DD}$	V

##### (3) 1/3 bias method

( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $V_{L4} (\text{MIN.}) \leq V_{DD} \leq 5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	$V_{L4}$		2.5		$V_{DD}$	V

## 33.7.2 Internal voltage boosting method

## (1) 1/3 bias method

(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
LCD output voltage variation range	V <sub>L1</sub>	C1 to C4 <sup>Note 1</sup> = 0.47 μF <sup>Note 2</sup>	VLCD = 04H	0.90	1.00	1.08	V
			VLCD = 05H	0.95	1.05	1.13	V
			VLCD = 06H	1.00	1.10	1.18	V
			VLCD = 07H	1.05	1.15	1.23	V
			VLCD = 08H	1.10	1.20	1.28	V
			VLCD = 09H	1.15	1.25	1.33	V
			VLCD = 0AH	1.20	1.30	1.38	V
			VLCD = 0BH	1.25	1.35	1.43	V
			VLCD = 0CH	1.30	1.40	1.48	V
			VLCD = 0DH	1.35	1.45	1.53	V
			VLCD = 0EH	1.40	1.50	1.58	V
			VLCD = 0FH	1.45	1.55	1.63	V
			VLCD = 10H	1.50	1.60	1.68	V
			VLCD = 11H	1.55	1.65	1.73	V
VLCD = 12H	1.60	1.70	1.78	V			
VLCD = 13H	1.65	1.75	1.83	V			
Doubler output voltage	V <sub>L2</sub>	C1 to C4 <sup>Note 1</sup> = 0.47 μF	2 V <sub>L1</sub> - 0.10	2 V <sub>L1</sub>	2 V <sub>L1</sub>	V	
Tripler output voltage	V <sub>L4</sub>	C1 to C4 <sup>Note 1</sup> = 0.47 μF	3 V <sub>L1</sub> - 0.15	3 V <sub>L1</sub>	3 V <sub>L1</sub>	V	
Reference voltage setup time <sup>Note 2</sup>	t <sub>VWAIT1</sub>		5			ms	
Voltage boost wait time <sup>Note 3</sup>	t <sub>VWAIT2</sub>	C1 to C4 <sup>Note 1</sup> = 0.47 μF	500			ms	

**Notes** 1. This is a capacitor that is connected between voltage pins used to drive the LCD.

C1: A capacitor connected between CAPH and CAPL

C2: A capacitor connected between V<sub>L1</sub> and GND

C3: A capacitor connected between V<sub>L2</sub> and GND

C4: A capacitor connected between V<sub>L4</sub> and GND

C1 = C2 = C3 = C4 = 0.47 μF ± 30%

- This is the time required to wait from when the reference voltage is specified by using the VLCD register (or when the internal voltage boosting method is selected (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default value reference voltage is used) until voltage boosting starts (VLCON = 1).
- This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).



**(2) 1/4 bias method****( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
LCD output voltage variation range	$V_{L1}$	C1 to C5 <sup>Note 1</sup> = $0.47\ \mu\text{F}$ <sup>Note 2</sup>	VLCD = 04H	0.90	1.00	1.08	V
			VLCD = 05H	0.95	1.05	1.13	V
			VLCD = 06H	1.00	1.10	1.18	V
			VLCD = 07H	1.05	1.15	1.23	V
			VLCD = 08H	1.10	1.20	1.28	V
			VLCD = 09H	1.15	1.25	1.33	V
			VLCD = 0AH	1.20	1.30	1.38	V
Doubler output voltage	$V_{L2}$	C1 to C5 <sup>Note 1</sup> = $0.47\ \mu\text{F}$	$2 V_{L1} - 0.08$	$2 V_{L1}$	$2 V_{L1}$	V	
Tripler output voltage	$V_{L3}$	C1 to C5 <sup>Note 1</sup> = $0.47\ \mu\text{F}$	$3 V_{L1} - 0.12$	$3 V_{L1}$	$3 V_{L1}$	V	
Quadruply output voltage	$V_{L4}$	C1 to C5 <sup>Note 1</sup> = $0.47\ \mu\text{F}$	$4 V_{L1} - 0.16$	$4 V_{L1}$	$4 V_{L1}$	V	
Reference voltage setup time <sup>Note 2</sup>	$t_{\text{WAIT1}}$		5			ms	
Voltage boost wait time <sup>Note 3</sup>	$t_{\text{WAIT2}}$	C1 to C5 <sup>Note 1</sup> = $0.47\ \mu\text{F}$	500			ms	

**Notes** 1. This is a capacitor that is connected between voltage pins used to drive the LCD.

C1: A capacitor connected between CAPH and CAPL

C2: A capacitor connected between  $V_{L1}$  and GND

C3: A capacitor connected between  $V_{L2}$  and GND

C4: A capacitor connected between  $V_{L3}$  and GND

C5: A capacitor connected between  $V_{L4}$  and GND

$C1 = C2 = C3 = C4 = C5 = 0.47\ \mu\text{F} \pm 30\%$

- This is the time required to wait from when the reference voltage is specified by using the VLCD register (or when the internal voltage boosting method is selected (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default value reference voltage is used) until voltage boosting starts (VLCON = 1).
- This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

### 33.7.3 Capacitor split method

#### (1) 1/3 bias method

( $T_A = -40$  to  $+105^\circ\text{C}$ ,  $2.4\text{ V} \leq V_D \leq 5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$V_{L4}$ voltage	$V_{L4}$	C1 to C4 = $0.47\ \mu\text{F}$ <sup>Note 2</sup>		$V_{DD}$		V
$V_{L2}$ voltage	$V_{L2}$	C1 to C4 = $0.47\ \mu\text{F}$ <sup>Note 2</sup>	$\frac{2}{3} V_{L4} - 0.1$	$\frac{2}{3} V_{L4}$	$\frac{2}{3} V_{L4} + 0.1$	V
$V_{L1}$ voltage	$V_{L1}$	C1 to C4 = $0.47\ \mu\text{F}$ <sup>Note 2</sup>	$\frac{1}{3} V_{L4} - 0.1$	$\frac{1}{3} V_{L4}$	$\frac{1}{3} V_{L4} + 0.1$	V
Capacitor split wait time <sup>Note 1</sup>	$t_{\text{WAIT}}$		100			ms

**Notes** 1. This is the wait time from when voltage bucking is started ( $V_{LCON} = 1$ ) until display is enabled ( $LCDON = 1$ ).

2. This is a capacitor that is connected between voltage pins used to drive the LCD.

C1: A capacitor connected between CAPH and CAPL

C2: A capacitor connected between  $V_{L1}$  and GND

C3: A capacitor connected between  $V_{L2}$  and GND

C4: A capacitor connected between  $V_{L4}$  and GND

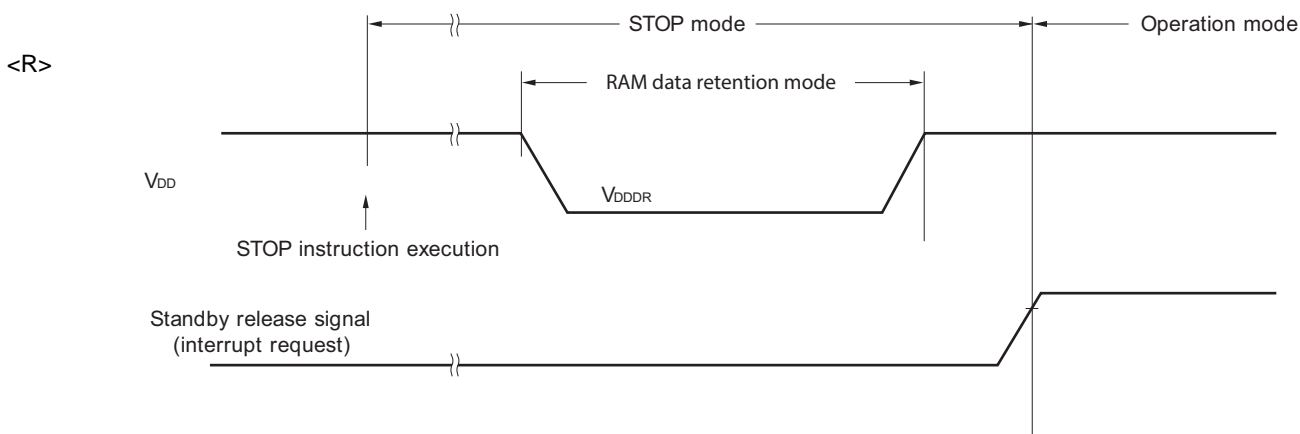
$C1 = C2 = C3 = C4 = 0.47\ \mu\text{F} \pm 30\%$

<R> **33.8 RAM Data Retention Characteristics**

(T<sub>A</sub> = -40 to +105°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V <sub>DDDR</sub>		1.44 <sup>Note</sup>		5.5	V

<R> **Note** This depends on the POR detection voltage. For a falling voltage, data in RAM are retained until the voltage reaches the level that triggers a POR reset but not once it reaches the level at which a POR reset is generated.



**33.9 Flash Memory Programming Characteristics**

(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency	f <sub>CLK</sub>	2.4 V ≤ V <sub>DD</sub> ≤ 5.5 V	1		24	MHz
Number of code flash rewrites <sup>Note 1, 2, 3</sup>	C <sub>erwr</sub>	Retained for 20 years T <sub>A</sub> = 85°C <sup>Note 4</sup>	1,000			Times
Number of data flash rewrites <sup>Note 1, 2, 3</sup>		Retained for 1 year T <sub>A</sub> = 25°C		1,000,000		
		Retained for 5 years T <sub>A</sub> = 85°C <sup>Note 4</sup>	100,000			
		Retained for 20 years T <sub>A</sub> = 85°C <sup>Note 4</sup>	10,000			

- Notes**
- 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. This characteristic indicates the flash memory characteristic and based on Renesas Electronics reliability test.
  4. This temperature is the average value at which data are retained.

**Remark** When updating data multiple times, use the flash memory as one for updating data.

**33.10 Dedicated Flash Memory Programmer Communication (UART)**

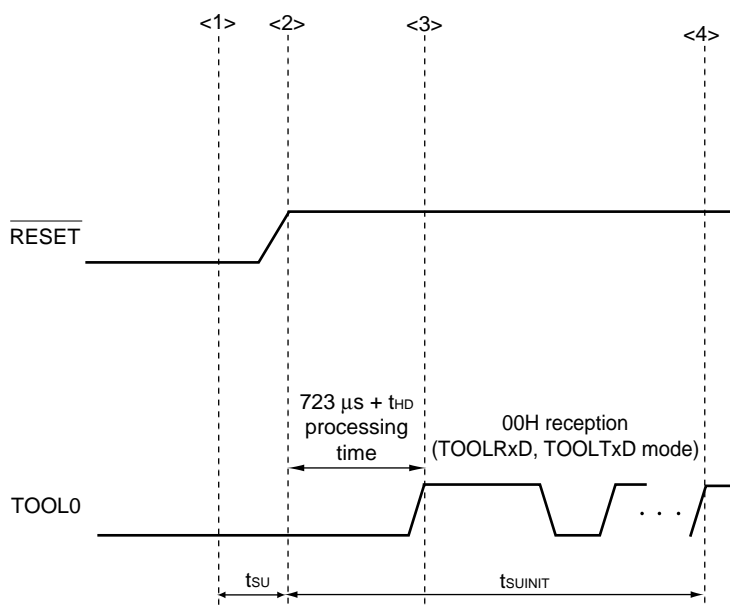
(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

### 33.11 Timing Specifications for Switching Flash Memory Programming Modes

(T<sub>A</sub> = -40 to +105°C, 2.4 V ≤ V<sub>DD</sub> ≤ 5.5 V, V<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Time to complete the communication for the initial setting after the external reset is released	t <sub>SUINIT</sub>	POR and LVD reset must be released before the external reset is released.			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	t <sub>SU</sub>	POR and LVD reset must be released before the external reset is released.	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)	t <sub>HD</sub>	POR and LVD reset must be released before the external reset is released.	1			ms



- <1> The low level is input to the TOOL0 pin.
- <2> The external reset is released (POR and LVD reset must be released before the external reset is released.).
- <3> The TOOL0 pin is set to the high level.
- <4> Setting of the flash memory programming mode by UART reception and completion the baud rate setting.

**Remark** t<sub>SUINIT</sub>: Communication for the initial setting must be completed within 100 ms after the external reset is released during this period.

t<sub>SU</sub>: Time to release the external reset after the TOOL0 pin is set to the low level

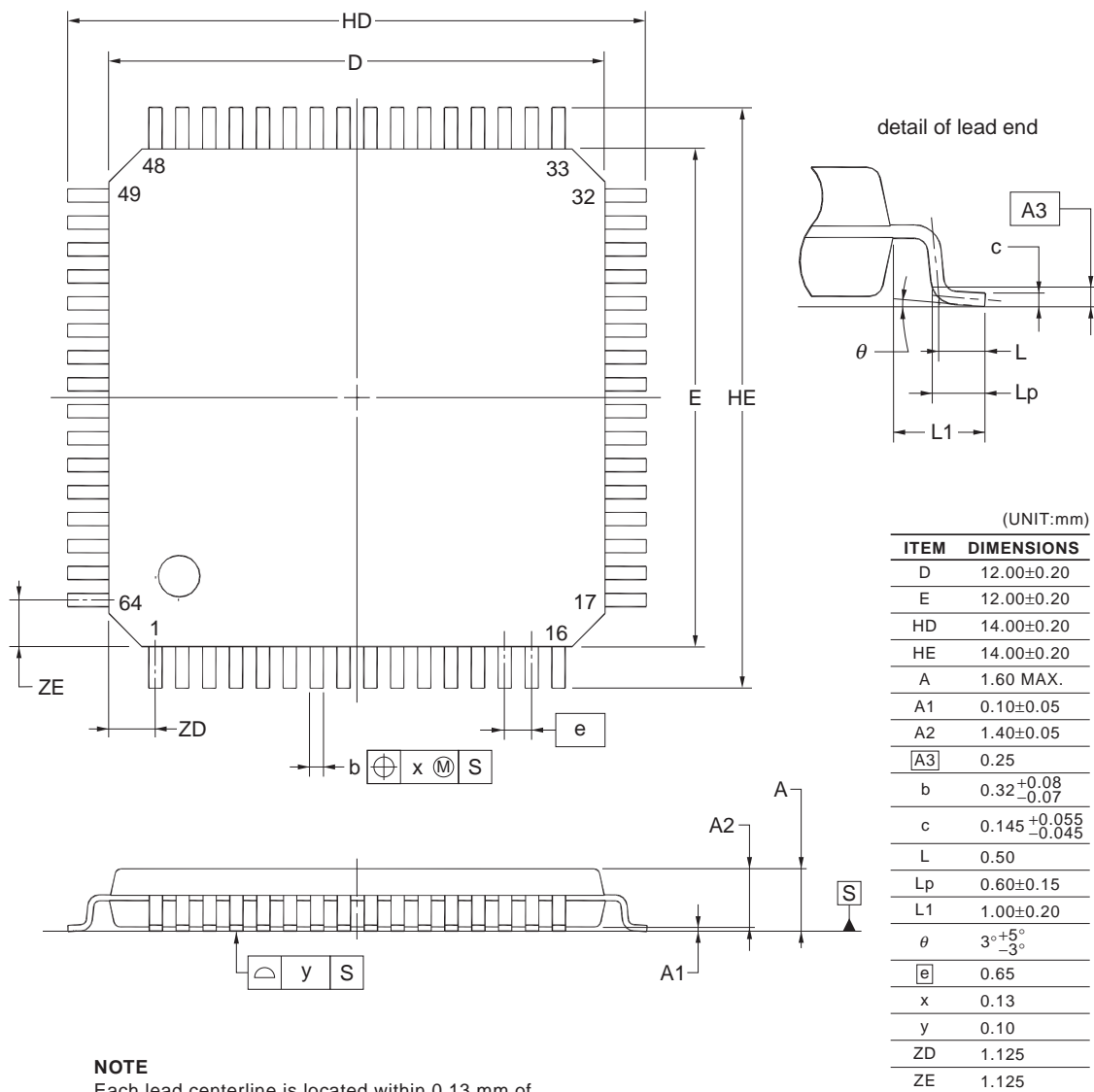
t<sub>HD</sub>: Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)

CHAPTER 34 PACKAGE DRAWINGS

34.1 64-pin Products

R5F10WLAFA, R5F10WLCAFA, R5F10WLDAFA, R5F10WLEAFA, R5F10WLFafa, R5F10WLGafa

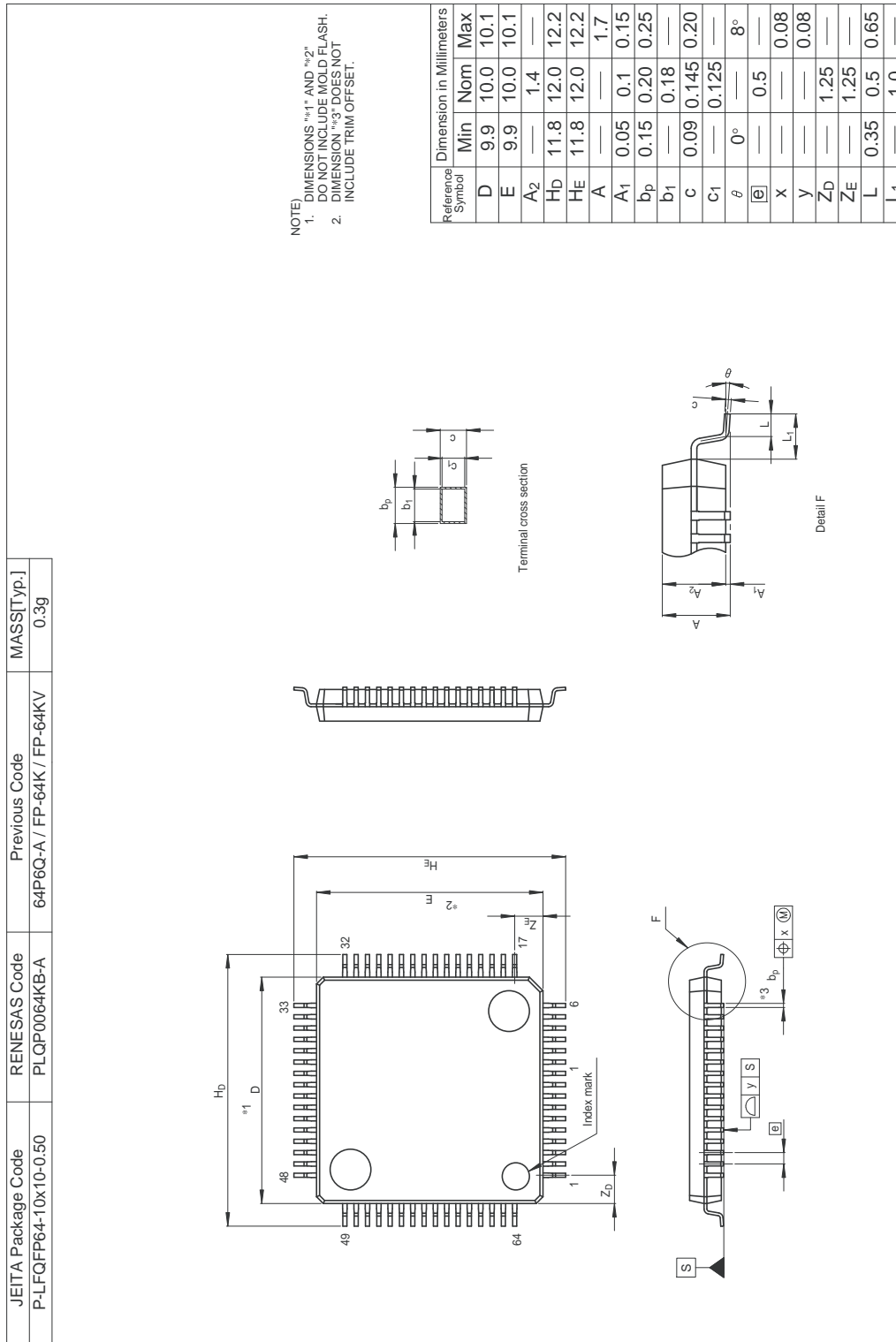
JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP64-12x12-0.65	PLQP0064JA-A	P64GK-65-UET-2	0.51



**NOTE**  
Each lead centerline is located within 0.13 mm of its true position at maximum material condition.

©2012 Renesas Electronics Corporation. All rights reserved.

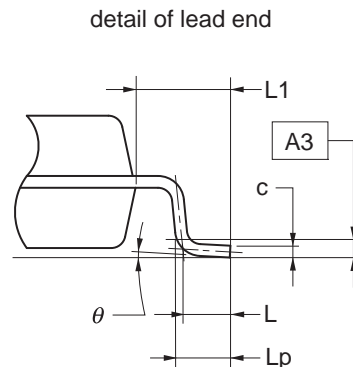
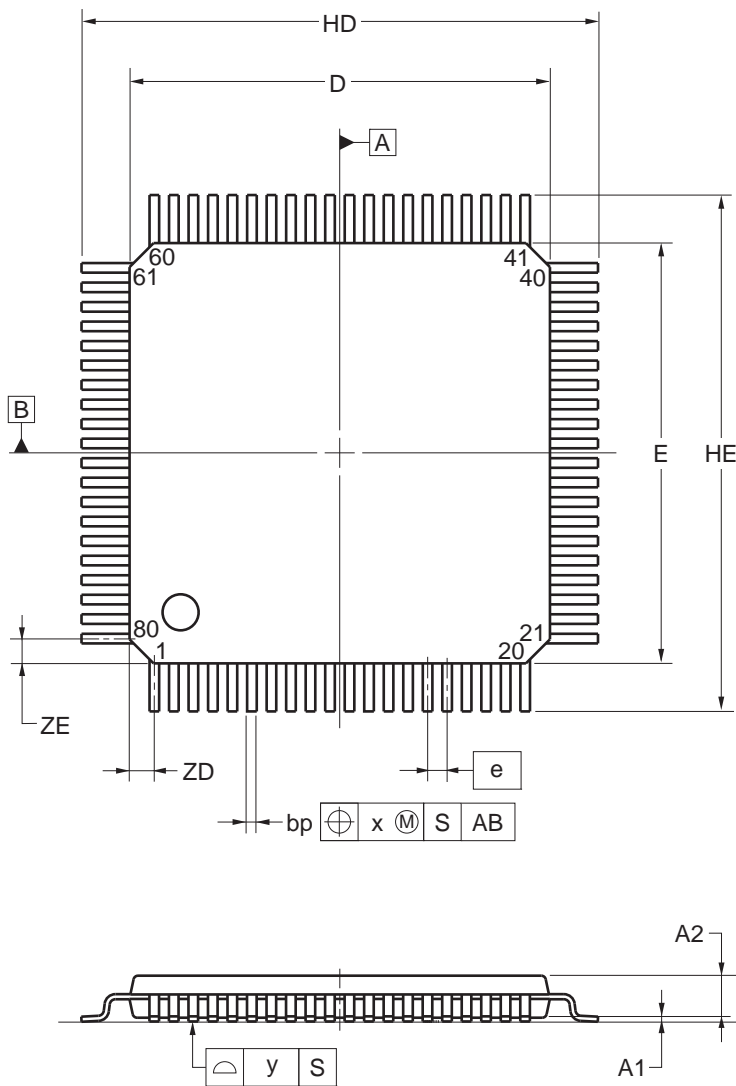
R5F10WLAAFB, R5F10WLCAFB, R5F10WLDAFB, R5F10WLEAFB, R5F10WLFafb, R5F10WLGafb, R5F10WLAGfb, R5F10WLCGfb, R5F10WLDGfb, R5F10WLEGfb, R5F10WLFgfb, R5F10WLGgfb



34.2 80-pin Products

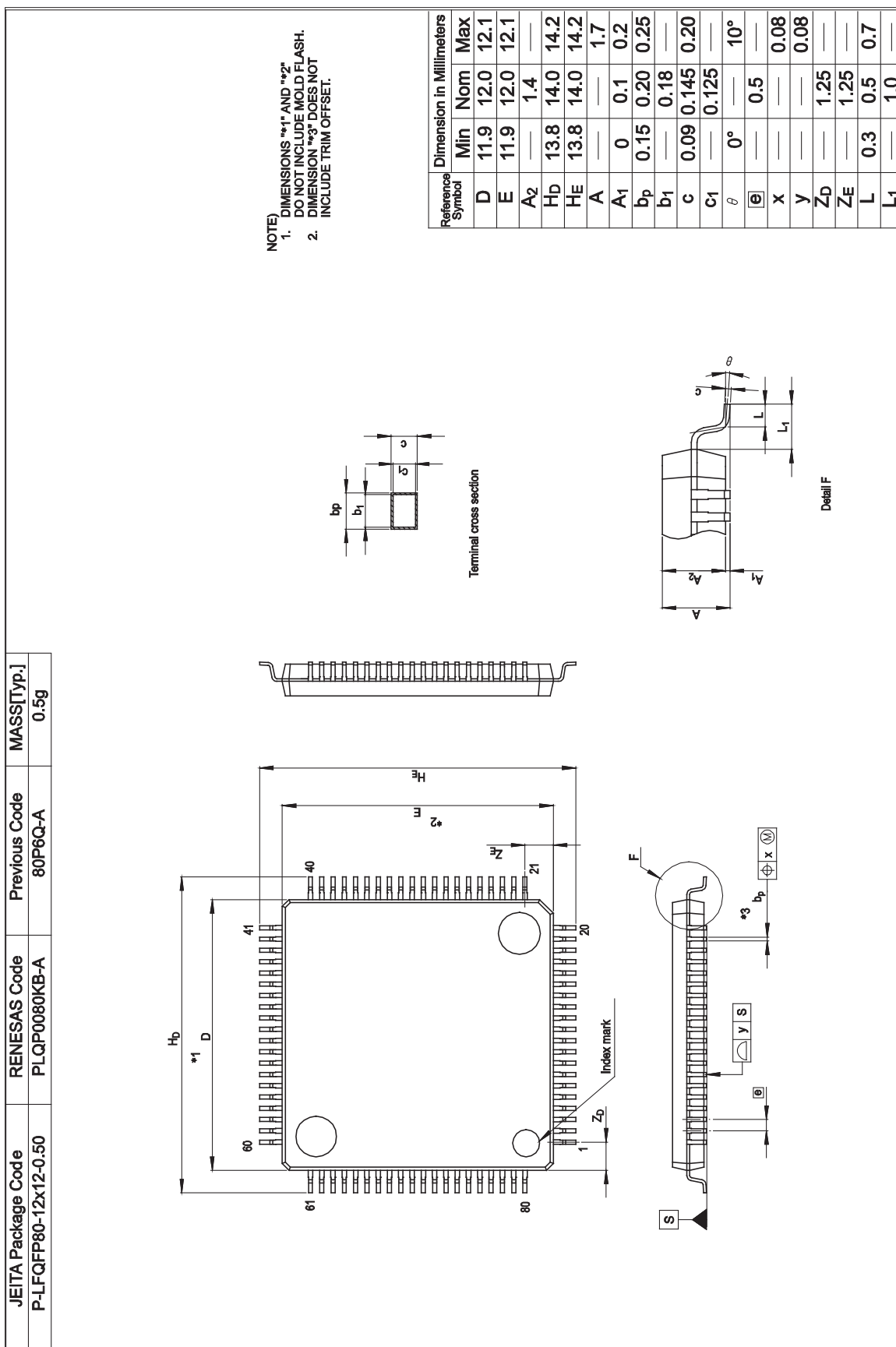
R5F10WMAAFA, R5F10WMCAFA, R5F10WMDAFA, R5F10WMEAFA, R5F10WMFAFA, R5F10WMGAFA

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP80-14x14-0.65	PLQP0080JB-E	P80GC-65-UBT-2	0.69



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.80	14.00	14.20
E	13.80	14.00	14.20
HD	17.00	17.20	17.40
HE	17.00	17.20	17.40
A	—	—	1.70
A1	0.05	0.125	0.20
A2	1.35	1.40	1.45
A3	—	0.25	—
bp	0.26	0.32	0.38
c	0.10	0.145	0.20
L	—	0.80	—
Lp	0.736	0.886	1.036
L1	1.40	1.60	1.80
	0°	3°	8°
e	—	0.65	—
x	—	—	0.13
y	—	—	0.10
ZD	—	0.825	—
ZE	—	0.825	—

R5F10WMAAFB, R5F10WMCAFB, R5F10WMDAFB, R5F10WMEAFB, R5F10WMFAFB, R5F10WMGAFB, R5F10WMAGFB, R5F10WMCGB, R5F10WMDGFB, R5F10WMEGFB, R5F10WMFGFB, R5F10WMGGFB





## APPENDIX A REVISION HISTORY

## A.1 Major Revisions in This Edition

(1/3)

Page	Description	Classification
<b>CHAPTER 1 OUTLINE</b>		
p.2	Modification of features of 16-bit timer and 16-bit timer KB20 (IH) in 1.1 Features	(c)
p.6	Addition of product name (RL78/L13) and description (Top View) in 1.3.1 64-pin products	(d)
p.7	Addition of product name (RL78/L13) and description (Top View) in 1.3.2 80-pin products	(d)
p.11	Modification of functional overview of main system clock in 1.6 Outline of Functions	(c)
<b>CHAPTER 2 PIN FUNCTIONS</b>		
p.15, 16	Modification of pin type of P40, P60, and P61, and modification of I/O column of P137 in 2.1.1 64-pin products	(a)
p.18, 19	Modification of pin type of P40, P60, and P61 in 2.1.2 80-pin products	(a)
p.29	Modification of Figure 2-4. Pin Block Diagram for Pin Type 4-3-3	(a)
p.30	Addition of Figure 2-5. Pin Block Diagram for Pin Type 7-1-3	(c)
p.34	Addition of caution in Figure 2-9 Pin Block Diagram for Pin Type 7-5-10	(c)
p.36	Addition of caution in Figure 2-11 Pin Block Diagram for Pin Type 7-6-4	(c)
p.39	Addition of caution in Figure 2-14 Pin Block Diagram for Pin Type 8-5-4	(c)
p.40	Addition of cautions 1 and 2 in Figure 2-15 Pin Block Diagram for Pin Type 8-5-10	(c)
p.41	Addition of cautions 1 and 2 in Figure 2-16 Pin Block Diagram for Pin Type 8-6-4	(c)
p.42	Addition of Figure 2-17 Pin Block Diagram for Pin Type 12-1-3	(c)
<b>CHAPTER 3 CPU ARCHITECTURE</b>		
p.53	Modification of vector table addresses in Table 3-3 Vector Table (1/2)	(a)
p.54	Modification of vector table addresses in Table 3-3 Vector Table (2/2)	(a)
p.60	Modification of addresses (00000H and 00001H) in (1) Program counter (PC) of 3.2.1 Control registers	(a)
<b>CHAPTER 4 PORT FUNCTIONS</b>		
p.123	Modification of description in (8) TKBO00 = 0/TKBO01-0, 1, 2 = 0 (setting when 16-bit timer KB20 is not used) of 4.5.2 Register settings for alternate function whose output function is not used	(a)
<b>CHAPTER 5 CLOCK GENERATOR</b>		
p.145	Modification of description in <2> High-speed on-chip oscillator of 5.1 Functions of Clock Generator	(c)
p.155	Modification of caution in Figure 5-4 Format of Clock Operation Status Control Register (CSC)	(c)
p.164	Modification of remark 1 in Figure 5-11 Format of High-Speed On-Chip Oscillator Trimming Register (HIOTRM)	(c)
p.169	Modification of description in 5.4.4 Low-speed on-chip oscillator	(a)
p.179	Modification of note in (6) and (8) of Table 5-3 CPU Clock Transition and SFR Setting Examples (3/5)	(b)
p.182	Modification of description in Table 5-4 Changing CPU Clock (1/2)	(c)
p.185	Modification of description in 5.6.7 Conditions before stopping clock oscillation	(c)
p.186	Modification of description in 5.7 Resonator and Oscillator Constants	(c)

**Remark** "Classification" in the above table classifies revisions as follows.

- (a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(2/3)

Page	Description	Classification
<b>CHAPTER 6 TIMER ARRAY UNIT</b>		
p.245	Modification of formula in (1) of 6.8.1 Operation as interval timer/square wave output	(b)
p.263	Modification of description in Figure 6-58 Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used	(a)
p.269	Modification of caution in 6.9.1 Operation as one-shot pulse output function	(b)
<b>CHAPTER 7 16-BIT TIMER KB20</b>		
p.295	Modification of description and table, and addition of caution 2	(c)
p.314	Modification of caution 3 in 7.3.16 16-bit timer KB2 counter restart select register (ELSELRn)	(b)
p.325	Modification of description in 7.3.24 Registers controlling port functions of 16-bit timer KB2 output pins	(a)
p.369	Modification of description in 7.5.4 PWM output function for IH control	(c)
<b>CHAPTER 12 A/D CONVERTER</b>		
p.476	Modification of description in Figure 12-27 Setting up Test Mode	(c)
p.479	Modification of description in Figure 12-31 Flowchart for Setting up SNOOZE Mode	(c)
<b>CHAPTER 14 SERIAL ARRAY UNIT</b>		
p.505	Modification of note 2 in 14.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)	(c)
p.515	Modification of caution 4 in Figure 14-9 Format of Serial Data Register mn (SDRmn)	(c)
p.525	Modification of pin name in Figure 14-18 Examples of Reverse Transmit Data	(a)
p.590	Modification of Figure 14-71 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)	(b)
p.591	Modification of Figure 14-72 Flowchart of SNOOZE Mode Operation (Once Startup)	(c)
p.592	Modification of Figure 14-73. Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)	(b)
p.593	Modification of Figure 14-74 Flowchart of SNOOZE Mode Operation (Continuous Startup)	(c)
p.616	Modification of description and addition of caution 5 in 14.6.3 SNOOZE mode function	(c)
p.618	Modification of Figure 14-90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)	(b)
p.619	Modification of Figure 14-91 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)	(b)
p.620	Modification of Figure 14-92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)	(c)
p.621	Modification of Figure 14-93 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)	(b)
p.622	Modification of Figure 14-94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)	(c)
p.634	Modification of Figure 14-101 Flowchart for LIN Reception	(c)
<b>CHAPTER 15 SERIAL INTERFACE IICA</b>		
p.676	Modification of description in 15.3.6 IICA low-level width setting register n (IICWLn)	(c)
p.695	Modification of formula in 15.5.14 Communication reservation	(b)
p.697	Modification of notes and remark in Figure 15-27 Communication Reservation Protocol	(b)
p.702	Addition of description in Figure 15-29 Master Operation in Multi-Master System (1/3)	(c)

**Remark** "Classification" in the above table classifies revisions as follows.

- (a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(3/3)

Page	Description	Classification
p.703	Modification of note and addition of description in Figure 15-29 Master Operation in Multi-Master System (2/3)	(c)
p.706	Addition of description in Figure 15-30 Slave Operation Flowchart (1)	(c)
<b>CHAPTER 19 INTERRUPT FUNCTIONS</b>		
p.843 to 845	Modification of vector table addresses in Tables 19-1 to 19 - 3 Interrupt Source List	(a)
<b>CHAPTER 22 RESET FUNCTION</b>		
All	Modification of addresses	(a)
p.892	Modification of address (00000H, 00001H) in Table 22-2 State of Hardware After Receiving a Reset Signal	(a)
p.895	Addition of description in Figure 22-6 Example of Procedure for Checking Reset Source	(c)
<b>CHAPTER 23 POWER-ON-RESET CIRCUIT</b>		
p.897	Modification of notes 3 and 4 in Figure 23-2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3)	(c)
p.899	Modification of note 3 in Figure 23-2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)	(c)
<b>CHAPTER 24 VOLTAGE DETECTOR</b>		
p.901	Modification of description in 24.1 Functions of Voltage Detector	(c)
<b>CHAPTER 27 OPTION BYTE</b>		
p.938	Modification of description in (3) 000C2H/010C2H of 27.1 Functions of Option Bytes	(c)
<b>CHAPTER 28 FLASH MEMORY</b>		
p.948	Modification of Table 28-1 Wiring Between RL78 Microcontroller and Dedicated Flash Memory Programmer	(a)
p.949	Addition of FLMD1 in Figure 28-2 Communication with Dedicated Flash Memory Programmer	(a)
p.950	Modification of signal name in Table 28-2 Pin Connection	(a)
<b>CHAPTER 29 ON-CHIP DEBUG FUNCTION</b>		
p.970	Modification of description and addition of note in Table 29-1 On-chip Debug Security ID	(c)
<b>CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to + 85°C)</b>		
p.999	Modification of description in Absolute Maximum Ratings (3/3)	(c)
p.1002	Modification of description in 32.3.1 Pin characteristics	(b)
p.1024	Modification of remark 3 in 32.5.1 (4) During communication at same potential (simplified I <sup>2</sup> C mode)	(b)
p.1057	Modification of the title and note, and addition of caution in 32.8 RAM Data Retention Characteristics	(b)
<b>CHAPTER 33 ELECTRICAL SPECIFICATIONS (G: INDUSTRIAL APPLICATIONS T<sub>A</sub> = -40 to + 105°C)</b>		
p.1059	Addition of Remark	(a)
p.1063	Modification of description in Absolute Maximum Ratings (T <sub>A</sub> = 25 °C) (3/3)	(b)
p.1066	Modification of description in 33.3.1 Pin characteristics	(b)
p.1085	Modification of remark 3 in 33.5.1 (4) During communication at same potential (simplified I <sup>2</sup> C mode)	(b)
p.1112	Modification of the title and note, and addition of caution in 33.8 RAM Data Retention Characteristics	(b)

**Remark** "Classification" in the above table classifies revisions as follows.

- (a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

### A.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/10)

Edition	Description	Chapter
Rev.2.00	Deletion of overscore over SCK** and ACK	Throughout
	Change of f <sub>EXT</sub> to f <sub>EXS</sub> , high accuracy RTC to RTC2, and high accuracy real-time clock to real-time clock 2	
	Modification of <b>1.1 Features</b>	<b>CHAPTER 1 OUTLINE</b>
	Modification of <b>1.2 List of Part Numbers</b>	
	Modification of <b>1.3.1 64-pin products</b>	
	Modification of <b>1.3.2 80-pin products</b>	
	Modification of <b>1.4 Pin Identification</b>	
	Modification of <b>1.5.1 64-pin products</b>	
	Modification of <b>1.5.2 80-pin products</b>	
	Modification of <b>1.6 Outline of Functions</b>	
	Modification of <b>2.1.1 64-pin products</b>	
	Modification of <b>2.1.2 80-pin products</b>	
	Modification of <b>2.2.1 With functions for each product</b>	
	Modification of <b>2.2.2 Description of Functions</b>	
	Modification of <b>2.3 Connection of Unused Pins</b>	
	Addition of <b>2.4 Block Diagrams of Pins</b>	
	Modification of <b>Note and Caution in Figure 3-1. Memory Map (R5F10WLA, R5F10WMA), Figure 3-2. Memory Map (R5F10WLC, R5F10WMC), Figure 3-3. Memory Map (R5F10WLD, R5F10WMD), Figure 3-4. Memory Map (R5F10WLE, R5F10WME), Figure 3-5. Memory Map (R5F10WLF, R5F10WMF), and Figure 3-6. Memory Map (R5F10WLG, R5F10WVG)</b>	<b>CHAPTER 3 CPU ARCHITECTURE</b>
	Modification of <b>Table 3-3. Vector Table (2/2)</b>	
	Modification of <b>3.1.3 Internal data memory space</b>	
	Modification of <b>3.1.6 Data memory addressing</b>	
	Modification of <b>3.2.1 (3) Stack pointer (SP)</b>	
	Modification of <b>3.2.2 General-purpose registers</b>	
	Modification of <b>3.2.3 ES and CS registers</b>	
	Modification of <b>3.2.4 Special function registers (SFRs)</b>	
	Modification of <b>Table 3-5. SFR List (2/5) and (3/5)</b>	
	Modification of <b>3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)</b>	
	Modification of <b>Table 3-6. Extended SFR (2nd SFR) List (2/9)</b>	
	Modification of <b>Figure 3-15. Outline of Relative Addressing</b>	
	Modification of <b>Figure 3-16. Example of CALL !addr20/BR !addr20</b>	
	Modification of <b>Figure 3-17. Example of CALL !addr16/BR !addr16</b>	
	Modification of <b>Figure 3-19. Outline of Register Direct Addressing</b>	
	Modification of <b>Figure 3-20. Outline of Implied Addressing</b>	
	Modification of <b>Figure 3-21. Outline of Register Addressing</b>	
	Modification of <b>3.4.3 Direct addressing</b>	

(2/10)

Edition	Description	Chapter	
Rev.2.00	Modification of <b>Figure 3-24. Outline of Short Direct Addressing</b>	<b>CHAPTER 3 CPU ARCHITECTURE</b>	
	Modification of <b>3.4.5 SFR addressing</b>		
	Modification of <b>Figure 3-26. Example of [DE], [HL]</b>		
	Modification of <b>Figure 3-27. Example of ES:[DE], ES:[HL]</b>		
	Modification of <b>3.4.7 Based addressing</b>		
	Modification of <b>Figure 3-35. Example of [HL+B], [HL+C]</b>		
	Modification of <b>Figure 3-36. Example of ES:[HL+B], ES:[HL+C]</b>		
	Modification of <b>3.4.9 Stack addressing</b>		
Rev.2.00	Modification of <b>Table 4-1 Port Configuration</b>	<b>CHAPTER 4 PORT FUNCTIONS</b>	
	Modification of <b>4.2.1 Port 0</b>		
	Modification of <b>4.2.2 Port 1</b>		
	Modification of <b>4.2.3 Port 2</b>		
	Modification of <b>4.2.4 Port 3</b>		
	Modification of <b>4.2.5 Port 4</b>		
	Modification of <b>4.2.6 Port 5</b>		
	Modification of <b>4.2.7 Port 6</b>		
	Modification of <b>4.2.8 Port 7</b>		
	Modification of <b>4.2.9 Port 12</b>		
	Modification of <b>4.2.10 Port 13</b>		
	Modification of <b>4.3 Registers Controlling Port Function</b>		
	Modification of <b>4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers</b>		
	Modification of <b>4.5 Register Settings When Using Alternate Function</b>		
	Rev.2.00		Modification of <b>5.1 (1) Main system clock</b>
Modification of <b>5.1 (2) Subsystem clock</b>			
Modification of <b>Table 5-1. Configuration of Clock Generator</b>			
Modification of <b>Figure 5-1. Block Diagram of Clock Generator</b>			
Modification of <b>5.3 Registers Controlling Clock Generator</b>			
Modification of <b>5.4 System Clock Oscillator</b>			
Modification of <b>5.5 Clock Generator Operation</b>			
Modification of <b>5.6 Controlling Clock</b>			
Addition of <b>5.7 Resonator and Oscillator Constants</b>			
Modification of <b>CHAPTER 6 TIMER ARRAY UNIT</b>			
Rev.2.00	Modification of <b>6.1 Functions of Timer Array Unit</b>	<b>CHAPTER 6 TIMER ARRAY UNIT</b>	
	Modification of <b>6.2 Configuration of Timer Array Unit</b>		
	Modification of <b>6.3 Registers Controlling Timer Array Unit</b>		
	Modification of <b>6.5 Operation of Counter</b>		
	Modification of <b>6.6 Channel Output (TOMn pin) Control</b>		
	Addition of <b>6.7 Timer Input (TIMn) Control</b>		
	Modification of <b>6.8 Independent Channel Operation Function of Timer Array Unit</b>		
	Modification of <b>6.9 Simultaneous Channel Operation Function of Timer Array Unit</b>		
	Modification of <b>7.3 Registers Controlling 16-bit Timer KB20</b>		<b>CHAPTER 7 16-BIT TIMER KB20</b>
	Modification of <b>7.4 Operation of 16-bit Timer KB20</b>		

(3/10)

Edition	Description	Chapter
Rev.2.00	Modification of 7.5 Option Functions of 16-bit Timer KB20	CHAPTER 7 16-BIT TIMER KB20
	Modification of 7.9 Cautions When Using 16-bit Timer KB20	
	Modification of 8.1 Functions of Real-time Clock 2	CHAPTER 8 REAL- TIME CLOCK 2
	Modification of 8.2 Configuration of Real-time Clock 2	
	Modification of 8.3 Registers Controlling Real-time Clock 2	
	Modification of 8.4 Real-time Clock 2 Operation	
	Modification of 9.2 Configuration of 12-bit Interval Timer	CHAPTER 9 12-BIT INTERVAL TIMER
	Modification of 9.3 Registers Controlling 12-bit Interval Timer	
	Modification of 9.4 12-bit Interval Timer Operation	
	Modification of 10.1 Functions of Clock Output/Buzzer Output Controller	CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER
	Modification of 10.2 Configuration of Clock Output/Buzzer Output Controller	
	Modification of 10.3 Registers Controlling Clock Output/Buzzer Output Controller	
	Modification of 10.4 Operations of Clock Output/Buzzer Output Controller	
	Modification of 10.5 Cautions of Clock Output/Buzzer Output Controller	
	Modification of 11.1 Functions of Watchdog Timer	CHAPTER 11 WATCHDOG TIMER
	Modification of 11.2 Configuration of Watchdog Timer	
	Modification of 11.4 Operation of Watchdog Timer	
	Modification of 12.1 Function of A/D Converter	CHAPTER 12 A/D CONVERTER
	Modification of 12.2 Configuration of A/D Converter	
	Modification of 12.3 Registers Used in A/D Converter	
	Modification of 12.4 A/D Converter Conversion Operations	
	Modification of 12.6 A/D Converter Operation Modes	
	Modification of 12.7 A/D Converter Setup Flowchart	
	Modification of 12.8 SNOOZE Mode Function	
	Modification of 12.9 How to Read A/D Converter Characteristics Table	
	Modification of 12.10 Cautions for A/D Converter	
	Addition of 13.1 Functions of Comparator	
	Addition of 13.2 Configuration of Comparator	
	Modification of 13.3 Registers Controlling Comparator	
	Modification of 13.4 Operation	
	Modification of 14.1 Functions of Serial Array Unit	CHAPTER 14 SERIAL ARRAY UNIT
	Modification of 14.2 Configuration of Serial Array Unit	
	Modification of 14.3 Registers Controlling Serial Array Unit	
	Modification of 14.5 Operation of 3-Wire Serial I/O (CSI00, CSI10) Communication	
	Modification of 14.6 Operation of UART (UART0 to UART3) Communication	
	Modification of 14.7 LIN Communication Operation	
	Modification of 14.8 Operation of Simplified I <sup>2</sup> C (IIC00, IIC10) Communication	
	Modification of 15.1 Functions of Serial Interface IICA	CHAPTER 15 SERIAL INTERFACE IICA
	Modification of 15.3 Registers Controlling Serial Interface IICA	
	Modification of 15.5 I <sup>2</sup> C Bus Definitions and Control Methods	
Modification of Table 16-1. Number of LCD Display Function Pins of Each Product	CHAPTER 16 LCD CONTROLLER/DRIVER	
Modification of 16.1 Functions of LCD Controller/Driver		

(4/10)

Edition	Description	Chapter
Rev.2.00	Modification of <b>16.2 Configuration of LCD Controller/Driver</b>	<b>CHAPTER 16 LCD CONTROLLER/DRIVER</b>
	Modification of <b>16.3 Registers Controlling LCD Controller/Driver</b>	
	Modification of <b>16.5 Selection of LCD Display Register</b>	
	Modification of <b>16.6 Setting the LCD Controller/Driver</b>	
	Modification of <b>16.7 Operation Stop Procedure</b>	
	Modification of <b>16.8 Supplying LCD Drive Voltages <math>V_{L1}</math>, <math>V_{L2}</math>, <math>V_{L3}</math>, and <math>V_{L4}</math></b>	
	Modification of <b>16.10 Display Modes</b>	
	Modification of <b>16.2 Configuration of LCD Controller/Driver</b>	
	Modification of <b>16.3 Registers Controlling LCD Controller/Driver</b>	
	Modification of <b>16.5 Selection of LCD Display Register</b>	
	Modification of <b>16.6 Setting the LCD Controller/Driver</b>	
	Modification of <b>16.7 Operation Stop Procedure</b>	
	Modification of <b>16.8 Supplying LCD Drive Voltages <math>V_{L1}</math>, <math>V_{L2}</math>, <math>V_{L3}</math>, and <math>V_{L4}</math></b>	
	Modification of <b>16.10 Display Modes</b>	
	Modification of <b>17.2 Configuration of Multiplier and Divider/Multiply-Accumulator</b>	<b>CHAPTER 17 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR</b>
	Modification of <b>17.3 Register Controlling Multiplier and Divider/Multiply-Accumulator</b>	
	Modification of <b>17.4 Operations of Multiplier and Divider/Multiply-Accumulator</b>	
	Modification of <b>18.1 Functions of DMA Controller</b>	<b>CHAPTER 18 DMA CONTROLLER</b>
	Modification of <b>18.5 Example of Setting of DMA Controller</b>	
	Modification of <b>19.2 Interrupt Sources and Configuration</b>	<b>CHAPTER 19 INTERRUPT FUNCTIONS</b>
	Modification of <b>19.3 Registers Controlling Interrupt Functions</b>	
	Modification of <b>19.4 Interrupt Servicing Operations</b>	
	Modification of <b>20.1 Functions of Key Interrupt</b>	<b>CHAPTER 20 KEY INTERRUPT FUNCTION</b>
	Modification of <b>20.3 Register Controlling Key Interrupt</b>	
	Modification of <b>21.2 Registers Controlling Standby Function</b>	<b>CHAPTER 21 STANDBY FUNCTION</b>
	Modification of <b>21.3 Standby Function Operation</b>	
	Modification of <b>CHAPTER 22 RESET FUNCTION</b>	<b>CHAPTER 22 RESET FUNCTION</b>
	Modification of <b>22.1 Timing of Reset Operation</b>	
	Modification of <b>22.2 States of Operation During Reset Periods</b>	
	Modification of <b>22.3 Register for Confirming Reset Source</b>	
	Modification of <b>23.1 Functions of Power-on-reset Circuit</b>	<b>CHAPTER 23 POWER-ON-RESET CIRCUIT</b>
	Modification of <b>23.3 Operation of Power-on-reset Circuit</b>	
	Modification of <b>24.1 Functions of Voltage Detector</b>	<b>CHAPTER 24 VOLTAGE DETECTOR</b>
	Modification of <b>24.2 Configuration of Voltage Detector</b>	
	Modification of <b>24.3 Registers Controlling Voltage Detector</b>	
	Modification of <b>24.4 Operation of Voltage Detector</b>	
	Modification of <b>24.5 Cautions for Voltage Detector</b>	
	Modification of <b>25.1 Overview of Safety Functions</b>	<b>CHAPTER 25 SAFETY FUNCTIONS</b>
	Modification of <b>25.2 Registers Used by Safety Functions</b>	
	Modification of <b>25.3 Operation of Safety Functions</b>	



(5/10)

Edition	Description	Chapter
Rev.2.00	Modification of <b>27.1 Functions of Option Bytes</b>	<b>CHAPTER 27 OPTION BYTE</b>
	Modification of <b>27.2 Format of User Option Byte</b>	
	Modification of <b>CHAPTER 28 FLASH MEMORY</b>	<b>CHAPTER 28 FLASH MEMORY</b>
	Modification of <b>28.1 Serial Programming Using Flash Memory Programmer</b>	
	Modification of <b>28.2 Serial Programming Using External Device (that Incorporates UART)</b>	
	Modification of <b>28.3 Connection of Pins on Board</b>	
	Modification of <b>28.4 Serial Programming Method</b>	
	Modification of <b>28.5 Self-Programming</b>	
	Modification of <b>28.6 Security Settings</b>	
	Modification of <b>28.7 Data Flash</b>	
Modification of <b>29.1 Connecting E1 On-chip Debugging Emulator</b>	<b>CHAPTER 29 ON-CHIP DEBUG FUNCTION</b>	
Modification of <b>31.1 Conventions Used in Operation List</b>	<b>CHAPTER 31 INSTRUCTION SET</b>	
Modification of <b>31.2 Operation List</b>		
Modification of <b>CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)</b>	<b>CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)</b>	
Modification of <b>32.1 Absolute Maximum Ratings</b>		
Modification of <b>32.2 Oscillator Characteristics</b>		
Modification of <b>32.3 DC Characteristics</b>		
Modification of <b>32.4 AC Characteristics</b>		
Modification of <b>32.5 Peripheral Functions Characteristics</b>		
Modification of <b>32.6 Analog Characteristics</b>		
Modification of <b>32.7 LCD Characteristics</b>		
Modification of <b>32.8 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics</b>		
Modification of <b>32.9 Flash Memory Programming Characteristics</b>		
Modification of <b>32.10 Dedicated Flash Memory Programmer Communication (UART)</b>		
Modification of <b>32.11 Timing Specifications for Switching Flash Memory Programming Modes</b>		
Modification of <b>CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)</b>	<b>CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)</b>	
Modification of <b>33.1 Absolute Maximum Ratings</b>		
Modification of <b>33.2 Oscillator Characteristics</b>		
Modification of <b>33.3 DC Characteristics</b>		
Modification of <b>33.4 AC Characteristics</b>		
Modification of <b>33.5 Peripheral Functions Characteristics</b>		
Modification of <b>33.6 Analog Characteristics</b>		
Modification of <b>33.7 LCD Characteristics</b>		
Modification of <b>33.8 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics</b>		
Modification of <b>33.9 Flash Memory Programming Characteristics</b>		
Modification of <b>33.10 Dedicated Flash Memory Programmer Communication (UART)</b>		
Modification of <b>33.11 Timing Specifications for Switching Flash Memory Programming Modes</b>		



(6/10)

Edition	Description	Chapter
Rev.2.00	Modification of <b>34.1 64-pin Products</b>	<b>CHAPTER 34</b>
	Modification of <b>34.2 80-pin Products</b>	<b>PACKAGE DRAWINGS</b>
Rev.1.00	Modification of segment signal output and operating ambient temperature in <b>1.1 Features</b>	<b>CHAPTER 1 OUTLINE</b>
	Modification of <b>1.2 Ordering Information</b>	
	Modification of <b>Figure 1-1. Part Number, Memory Size, and Package of RL78/L13</b>	
	Modification of segment signal output, power-on-reset circuit, and operating ambient temperature in <b>1.6 Outline of Functions</b>	
	Modification of segment signal output pins in <b>2.2.1 With functions for each product</b>	<b>CHAPTER 2 PIN FUNCTIONS</b>
	Modification of segment signal output pins in <b>2.2.2 Description of Functions</b>	
	Modification of P45/IVREF0 in <b>Table 2-2. Connection of Unused Pins (80-pin products)</b>	
	Modification of <b>3.1.2 Mirror area</b>	<b>CHAPTER 3 CPU ARCHITECTURE</b>
	Modification of <b>Figure 3-19. Configuration of ES and CS Registers</b>	
	Modification of <b>Table 3-6. Extended SFR (2nd SFR) List (8/10)</b>	
	Modification of <b>note</b> in <b>4.2.1 Port 0, 4.2.2 Port 1, 4.2.4 Port 3, 4.2.5 Port 4, 4.2.6 Port 5, 4.2.8 Port 7, and 4.2.10 Port 13</b>	<b>CHAPTER 4 PORT FUNCTIONS</b>
	Modification of P45 in <b>Table 4-8. Settings of Registers When Using Port 4</b>	
	Modification of <b>Figure 4-19. Block Diagram of P45</b>	
	Modification of <b>4.3 (9) LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)</b>	
	Modification of <b>Figure 4-43. Format of LCD port function registers 0 to 6 (PFSEG0 to PFSEG6) (80-pin products)</b>	
	Modification of P45 in <b>Table 4-16. Settings of Port Related Register When Using Alternate Function (4/7)</b>	
	Addition of <b>4.5.3 P45 (port that serves as an analog input pin (IVREF0))</b>	
	Modification of <b>Table 4-24. Handling of Unused Alternate Functions</b>	
	Modification of <b>Figure 5-1. Block Diagram of Clock Generator</b>	<b>CHAPTER 5 CLOCK GENERATOR</b>
	Modification of <b>note</b> in <b>Figure 5-8. Format of Peripheral Enable Register 1 (PER1)</b>	
	Modification of <b>5.3 (8) High-speed on-chip oscillator frequency select register (HOCODIV)</b>	
	Modification of <b>cautions 1 to 3</b> in <b>Figure 5-10. Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV)</b>	
	Modification of <b>Figure 5-14. Clock Generator Operation When Power Supply Voltage Is Turned On</b>	
	Modification of [Option byte setting] in <b>5.6.1 Example of setting high-speed on-chip oscillator</b>	
	Modification of <b>Figure 5-15. CPU Clock Status Transition</b>	
	Modification of <b>Table 5-3. CPU Clock Transition and SFR Setting Examples (2/5) and (3/5)</b>	
	Modification of <b>Figure 6-1. Entire Configuration of Timer Array Unit</b>	<b>CHAPTER 6 TIMER ARRAY UNIT</b>
	Modification of <b>Figure 6-2. Internal Block Diagram of Channels 0, 2, 4, 6 of Timer Array Unit to Figure 6-6. Internal Block Diagram of Channels 7 of Timer Array Unit</b>	
	Modification of <b>6.2 (2) Timer data register mn (TDRmn)</b>	
	Addition of <b>caution 1</b> to <b>Figure 6-23. Format of Input Switch Control Register (ISC)</b>	

(7/10)

Edition	Description	Chapter
Rev.1.00	Modification of <b>Figure 7-1. Block Diagram of 16-bit Timer KB20</b>	<b>CHAPTER 7 16-BIT TIMER KB20</b>
	Modification of <b>notes 2 and 3 in Figure 7-4. Format of Timer Clock Select Register (TKBPSCS0)</b>	
	Modification of <b>Figure 7-5. Format of 16-bit Timer KB2 Operation Control Register 00 (TKBCTL00)</b>	
	Modification of <b>Figure 7-8. Format of 16-bit Timer KB2 Output Control Register 01 (TKBIOC01)</b>	
	Modification of <b>Figure 7-9. Format of 16-bit Timer KB2 Flag Register 0 (TKBFLG0)</b>	
	Modification of <b>Figure 7-11. Format of 16-bit Timer KB2 Flag Clear Trigger Register 0 (TKBCLR0)</b>	
	Modification of <b>Figure 7-21. Format of Forced Output Stop Function Control Register 0 (TKBPACTL00)</b>	
	Modification of <b>Figure 7-22. Format of Forced Output Stop Function Control Register 1 (TKBPACTL01)</b>	
	Modification of <b>Figure 7-23. Format of Forced Output Stop Function Control Register 2 (TKBPACTL02)</b>	
	Modification of <b>Figure 7-42. Timing Sample for Standalone Mode (Period Controlled by TKBCR00) (When default value of output is low level (TKBTOD0p = 0) and active level is high level (TKBTOL0p = 0))</b>	
	Modification of <b>Figure 7-44. Batch Overwrite Function: Figure of Timing of Buffer Updating during Counting Operation</b>	
	Modification of <b>Figure 7-46. Batch Overwrite Function: Figure of Standalone Operation during Period Controlled by External Trigger Input and Timing of Buffer Updating during Counting Operation (TKBTSE0 bit set to 1)</b>	
	Modification of <b>Figure 7-47. Batch Overwrite Function: Figure of Standalone Operation during Period Controlled by External Trigger Input and Timing of Buffer Updating during Counting Operation (TKBTSE0 bit cleared to 0)</b>	
	Modification of <b>Figure 7-51. Figure of Timing of Interleave PFC Mode (Operation for Status No. 5: INTO Input Detected (for the first time))</b>	
	Modification of <b>table in 7.5 Option Functions of 16-bit Timer KB20</b>	
	Modification of <b>Figure 7-59. Figure of Waveform at Dithering Operation and Figure 7-61. Figure of Waveform at Dithering Operation (When TKBCR03 = TKBCR00 + 1)</b>	
	Modification of <b>7.5.4 PWM Output Function for IH Control</b>	
	Modification of <b>7.6.1 Forced Output Stop Function 1 and 2</b>	
	Modification of <b>7.7.1 I/O Setting for Forced Output Stop Function 1</b>	
	Modification of <b>7.7.2 (1) TKBPAHCM0n1, TKBPAHCM0n0 Bits = 00 to (4) TKBPAHCM0n1, TKBPAHCM0n0 bits = 11</b>	
Modification of <b>7.7.3 (1) Setting TKBPAHCM0n = 10 and (2) Setting TKBPAHCM0n = 11</b>		
Modification of <b>7.8.1 I/O Setting for Forced Output Stop Function 2</b>		
Modification of <b>7.9 Cautions When Using 16-Bit Timer KB20</b>		
Modification of <b>Figure 8-1. High Accuracy Real-time Clock Diagram</b>	<b>CHAPTER 8 HIGH ACCURACY REAL- TIME CLOCK</b>	
Modification of <b>note 1 in 8.3 Registers Controlling High Accuracy Real-time Clock</b>		
Modification of <b>[Calculating the target correction value] in 8.4.8 Example of watch error correction of high accuracy real-time clock</b>		

(8/10)

Edition	Description	Chapter	
Rev.1.00	Modification of <b>12.2 (9) AV<sub>REFP</sub> pin</b>	<b>CHAPTER 12 A/D CONVERTER</b>	
	Modification of <b>cautions 1 and 3 in Figure 12-3. Format of A/D Converter Mode Register 0 (ADM0)</b>		
	Modification of <b>caution 1 in Table 12-3. A/D Conversion Time Selection</b>		
	Modification of <b>Figure 12-5. A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)</b>		
	Modification of <b>Figure 12-18. Conversion Operation of A/D Converter (Software Trigger Mode)</b>		
	Modification of <b>Figure 12-20. Example of Software Trigger Mode (Sequential Conversion Mode) Operation Timing to Figure 12-25. Example of Hardware Trigger Wait Mode (One-Shot Conversion Mode) Operation Timing</b>		
Rev.1.00	Modification of <b>Table 13-1. Comparator Specifications</b>	<b>CHAPTER 13 COMPARATOR</b>	
	Modification of <b>Figure 13-1. Comparator Block Diagram</b>		
	Modification of <b>note 4 and addition of notes 5 to 7 in Figure 13-3. Format of Comparator Mode Setting Register (COMPMDR)</b>		
	Addition of <b>13.3 (6) LCD Port Function Register 3 (PFSEG3)</b>		
	Modification of <b>Table 13-4. Procedure for Setting Comparator Associated Registers</b>		
	Modification of <b>Figure 13-12 Comparator i (i = 0 or 1) Digital Filter and Edge Detection Configuration</b>		
	Addition of <b>&lt;1&gt; to 13.4.3 Comparator i Output (i = 0 or 1)</b>		
	Addition of <b>caution to 13.4.4 Stopping or Supplying Comparator Clock</b>		
	Modification of Maximum transfer rate to <b>14.1.1 3-wire serial I/O (CSI00, CSI10) and 14.5 Operation of 3-Wire Serial I/O (CSI00, CSI10) Communication</b>		<b>CHAPTER 14 SERIAL ARRAY UNIT</b>
	Modification of <b>Figure 14-1. Block Diagram of Serial Array Unit 0 and Figure 14-2. Block Diagram of Serial Array Unit 1 (80-pin products)</b>		
	Modification of <b>caution 2 to Figure 14-19. Format of Input Switch Control Register (ISC)</b>		
	Modification of transfer rate to <b>14.5.1 Master transmission, 14.5.2 Master reception and 14.5.3 Master transmission/reception</b>		
Addition of <b>caution 3 to 14.6.3 SNOOZE mode function</b>			
Addition of <b>caution 3 to 14.7.2 LIN reception</b>			
Rev.1.00	Modification of <b>Figure 15-1. Block Diagram of Serial Interface IICA0</b>	<b>CHAPTER 15 SERIAL INTERFACE IICA</b>	
	Modification of <b>Table 16-1. Number of LCD Display Function Pins of Each Product</b>		<b>CHAPTER 16 LCD CONTROLLER/DRIVER</b>
	Modification of <b>(4) in 16.1 Functions of LCD Controller/Driver</b>		
	Modification of <b>Table 16-2. Maximum Number of Pixels</b>		
	Modification of <b>Table 16-4. Combinations of Display Waveform, Time Slices, and Bias Method</b>		
	Addition of <b>remark to Figure 16-5. Format of LCD Clock Control Register 0 (LCDC0) (1/2)</b>		
	Modification of <b>caution 2 and addition of caution 3 in Figure 16-5. Format of LCD Clock Control Register 0 (LCDC0) (2/2)</b>		
	Modification of <b>Figure 16-7. Format of LCD Boost Level Control Register (VLCD)</b>		
	Modification of <b>16.3 (8) LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)</b>		
	Modification of <b>Figure 16-11. Format of LCD Port Function Registers 0 to 6 (80-pin Products)</b>		

(9/10)

Edition	Description	Chapter
Rev.1.00	Modification of <b>Table 16-11. Relationship Between LCD Display Data Register Contents and Segment/Common Outputs</b>	<b>CHAPTER 16 LCD CONTROLLER/DRIVER</b>
	Modification of <b>caution</b> in <b>Figure 16-23. Examples of LCD Drive Power Connections (External Resistance Division Method)</b>	
	Modification of <example of calculation of LCD frame frequency> in <b>Figure 16-26. Common Signal Waveforms</b>	
	Modification of <b>Figure 16-27. Voltages and Phases of Common and Segment Signals (3/3)</b>	
	Modification of <b>Figure 16-40. Four-Time-Slice LCD Drive Waveform Examples Between SEG12 and Each Common Signals (1/3 Bias Method) (2/2)</b>	
	Modification of <b>Figure 16-43. Eight-Time-Slice LCD Drive Waveform Examples Between SEG4 and Each Common Signals (1/4 Bias Method)</b>	
	Modification of <b>Figure 17-9. Timing Diagram of Multiply-Accumulation (signed) Operation</b> ( $2 \times 3 + (-4) = 2 \rightarrow 32767 \times (-1) + (-2147483647) = -2147516414$ (overflow occurs.))	<b>CHAPTER 17 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR</b>
	Modification of <b>18.6 (6) Operation if instructions for accessing the data flash area</b>	<b>CHAPTER 18 DMA CONTROLLER</b>
	Modification of <b>caution 1</b> in <b>Figure 20-3. Format of Key Return Control Register (KRM0)</b>	<b>CHAPTER 20 KEY INTERRUPT FUNCTION</b>
	Modification of <b>20.3 (3) Key return flag register (KRF)</b>	
	Modification of <b>caution 1</b> in <b>Figure 20-4. Format of Key Return Flag Register (KRF)</b>	
	Modification of <b>Table 21-1. Operating Statuses in HALT Mode (2/2)</b>	<b>CHAPTER 21 STANDBY FUNCTION</b>
	Modification of description and addition of <b>caution 1</b> in this page	<b>CHAPTER 22 RESET FUNCTION</b>
	Modification of <b>23.1 Functions of Power-on-reset Circuit</b>	<b>CHAPTER 23 POWER-ON-RESET CIRCUIT</b>
	Modification of <b>23.3 Operation of Power-on-reset Circuit</b>	
	Modification of <b>Figure 23-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector</b>	
	Addition of <b>caution</b> to <b>Table 24-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H) (1/2)</b>	<b>CHAPTER 24 VOLTAGE DETECTOR</b>
	Addition of <b>cautions 1 and 2</b> to <b>Table 24-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H) (1/2)</b>	
	Modification of <b>Figure 24-5. Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)</b>	
	Modification of <b>Figure 24-6. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0)</b>	
	Addition of <b>(3)</b> and <b>(4)</b> to <b>24.5 Cautions for Voltage Detector</b>	
	Modification of <b>Figure 25-1. Format of Flash Memory CRC Control Register (CRC0CTL)</b>	
	Modification of <b>Figure 25-10. Invalid access detection area</b>	
	Addition of <b>cautions 2 to 5</b> to <b>Figure 25-17. Format of Port Mode Select Register (PMS)</b>	
	Addition of <b>caution</b> to <b>28.4.1 Data flash overview</b>	<b>CHAPTER 28 FLASH MEMORY</b>
	Modification of <b>cautions 2 and 3</b> in <b>28.7 Flash Memory Programming by Self-Programming</b>	
	Modification of <b>28.7.2 Flash shield window function</b>	
	Modification of <b>Table 28-14. Relationship between Flash Shield Window Function Setting/Change Methods and Commands</b>	

(10/10)

Edition	Description	Chapter
Rev.1.00	Modification of throughout the document	<b>CHAPTER 32 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +85°C)</b>
	Modification of throughout the document	<b>CHAPTER 33 ELECTRICAL SPECIFICATIONS (T<sub>A</sub> = -40 to +105°C)</b>
	Addition of part number and package drawings are updated.	<b>CHAPTER 34 PACKAGE DRAWINGS</b>

---

RL78/L13 User's Manual: Hardware

Publication Date: Rev.2.10 Aug 12, 2016

Published by: Renesas Electronics Corporation

---

**SALES OFFICES**

Renesas Electronics Corporation

<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.**Renesas Electronics America Inc.**2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130**Renesas Electronics Canada Limited**9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3  
Tel: +1-905-237-2004**Renesas Electronics Europe Limited**Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K  
Tel: +44-1628-585-100, Fax: +44-1628-585-900**Renesas Electronics Europe GmbH**Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-6503-0, Fax: +49-211-6503-1327**Renesas Electronics (China) Co., Ltd.**Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679**Renesas Electronics (Shanghai) Co., Ltd.**Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333  
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999**Renesas Electronics Hong Kong Limited**Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2265-6688, Fax: +852 2886-9022**Renesas Electronics Taiwan Co., Ltd.**13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670**Renesas Electronics Singapore Pte. Ltd.**80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949  
Tel: +65-6213-0200, Fax: +65-6213-0300**Renesas Electronics Malaysia Sdn.Bhd.**Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510**Renesas Electronics India Pvt. Ltd.**No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India  
Tel: +91-80-67208700, Fax: +91-80-67208777**Renesas Electronics Korea Co., Ltd.**12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

RL78/L13