User's Manual

16

78K0R/Lx3-M

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.1.00 Mar 2012

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

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 78K0R/Lx3-M microcontrollers and design and develop application systems and programs for these devices.
The target products are as follows.

- 78K0R/LG3-M: μPD78F8070

Purpose

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

Organization

The manual for the 78K0R/Lx3-M microcontrollers is separated into two parts: this manual and the instructions edition (common to the 78K0R Microcontroller Series).

78K0R/Lx3-M Preliminary User's Manual (This Manual)	**78K0R Microcontrollers User's Manual Instructions**
• 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 RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R.
- To know details of the 78K0R Series instructions:
  → Refer to the separate document **78K0R Microcontrollers Instructions User's Manual (R01US0029E)**.

Conventions

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

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

Documents Related to Devices

Document Name	Document No.
78K0R/Lx3-M User's Manual	This manual
78K0R Microcontrollers Instructions User's Manual	R01US0029E

Documents Related to Development Tools (Software) (User's Manuals)

Document Name		Document No.
CC78K0R Ver. 2.00 C Compiler	Operation	U18549E
	Language	U18548E
RA78K0R Ver. 1.20 Assembler Package	Operation	U18547E
	Language	U18546E
SM+ System Simulator	Operation	U18601E
	User Open Interface	U18212E
PM+ Ver.6.30		U18416E
ID78K0R-QB Ver. 3.20 Integrated Debugger	Operation	U17839E
CubeSuite+ Integrated Development Environment User's Manual	Start	R20UT0727E
	78K0R Design	R20UT0547E
	RL78, 78K0R Coding	R20UT0729E
	RL78, 78K0R Build	R20UT0730E
	78K0R Debug	R20UT0732E
	Analysis	R20UT0735E
	Message	R20UT0736E

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

Documents Related to Flash Memory Programming (User's Manuals)

Document Name	Document No.
PG-FP5 Flash Memory Programmer	R20UT0008E
QB-Programmer Programming GUI, Operation	U18527E

Other Documents

Document Name	Document No.
RENESAS MICROCOMPUTER GENERAL CATALOG	R01CS0001E
Semiconductor Package Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

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.

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

CONTENTS

CHAPTER 1 OUTLINE

The 78K0R/Lx3-M microcontrollers are 16-bit single-chip microcontrollers that include the 78K0R CPU core and peripheral functions such as ROM/RAM, LCD controller/driver, 10-bit successive approximation type A/D converter, multifunctional serial interfaces, multifunctional timers, real-time counter, and watchdog timer, as well as the 24-bit ΔΣ-type A/D converter, power calculation circuit, power quality measurement circuit, and digital frequency conversion circuit that are used for power measurement.

1.1 Features

- Minimum instruction execution time can be changed from high speed (0.05 μs: @ 20 MHz operation with high-speed system clock) to ultra low-speed (61 μs: @ 32.768 kHz operation with subsystem clock)
- General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)
- ROM, RAM capacities
  - Program Memory (ROM): 128 KB
  - Data Memory (RAM): 7 KB
- On-chip internal high-speed oscillation clock
  - 8 MHz internal high-speed oscillation clock: 8 MHz ± 5 %
  - 1 MHz internal high-speed oscillation clock: 1 MHz ± 13 %
- On-chip single-power-supply flash memory (with prohibition of chip erase/block erase/writing function)
- Self-programming (with boot swap function/flash shield window function)
- On-chip debug function
- On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)
- On-chip watchdog timer (can operate on dedicated internal low-speed oscillation clock)
- On-chip multiplier/divider (16 bits × 16 bits, 32 bits ÷ 32 bits)
- On-chip BCD adjustment
- I/O ports: 45 (N-ch open drain: 2)
- Timers
  - 16-bit timer: 12 channels (input: 3, output: 3)
  - Watchdog timer: 1 channel
  - Real-time counter: 1 channel (output: 1)
  - Real-time counter 2: 1 channel

- Serial interface
  UART: 1 channel
  CSI: 1 channel/UART: 1 channel/simplified I^2C: 1 channel
  UART (LIN-bus supported): 1 channel
  Multimaster I^2C: 1 channel
- 10-bit successive approximation type A/D converter: 2 channels
- 24-bit ΔΣ-type A/D converter: 4 channels
- LCD controller/driver (Internal voltage boosting method, capacitor split method, and external resistance division method are switchable)
  - Segment signal output: 40
  - Common signal output: 4
- DMA controller: 2 channels
- On-chip power calculation circuit
- On-chip power quality measurement circuit
- On-chip digital frequency conversion circuit
- Power supply voltage: VDD = 1.8 to 3.6 V
- Operating ambient temperature: TA = −40 to +85°C

1.2 Ordering Information

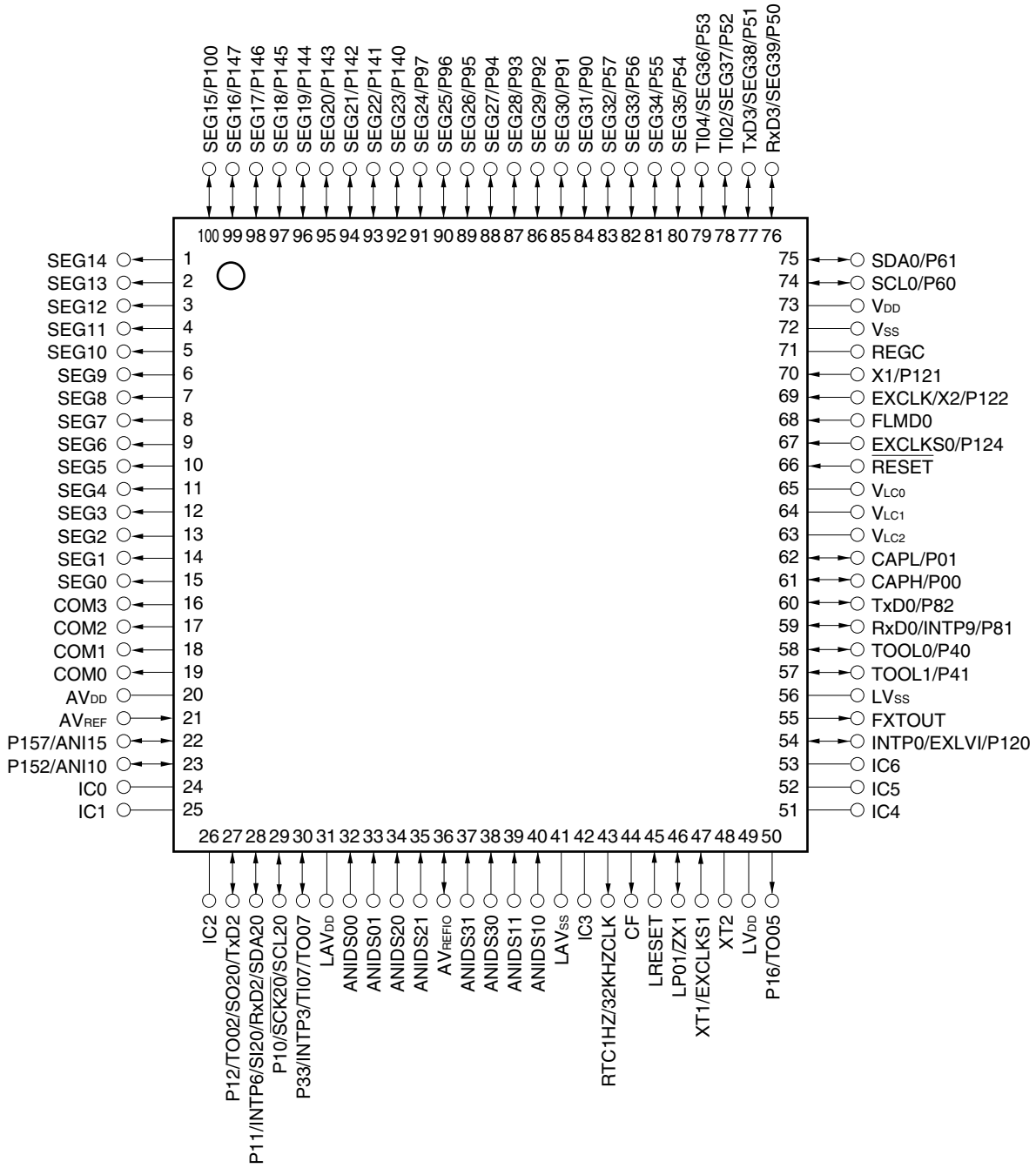
- **Flash memory version (Lead-free products)**

78K0R/Lx3-M microcontrollers	Package	Part Number
78K0R/LG3-M	100-pin plastic LQFP (fine pitch) (14 × 14)	μPD78F8070GC-UEU-AX

Caution The 78K0R/Lx3-M 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.

1.3 Pin Configuration (Top View)

- 100-pin plastic LQFP (fine pitch) (14 × 14)



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

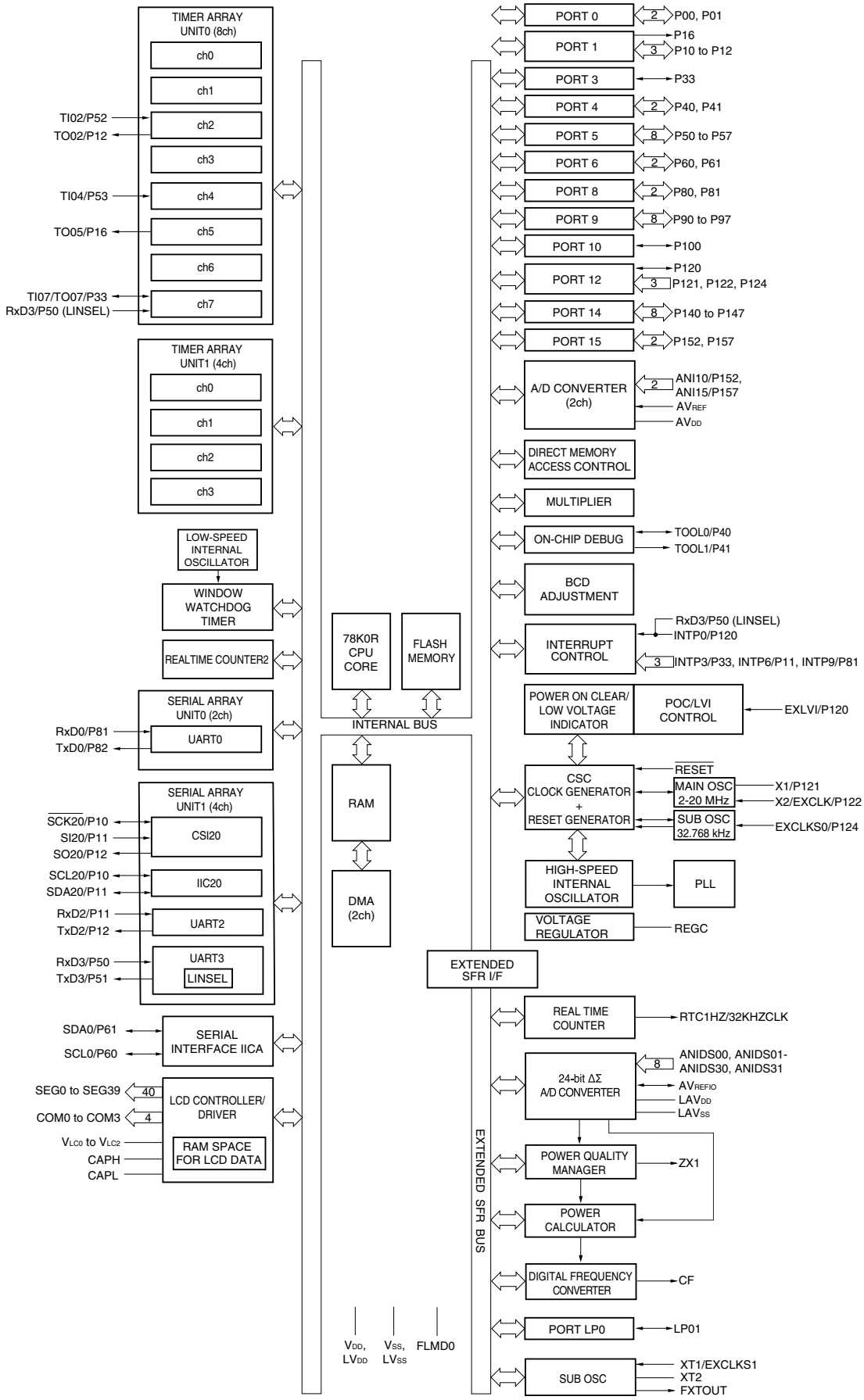
<R> **2. Perform the following connections as noise countermeasure.**

- **Short pins VSS and LVSS pin using wiring that is as short as possible and as thick as possible.**
- **Connect the $\overline{\text{RESET}}$ pin to GND via a bypass capacitor (capacity: approx. 47 μF).**
- **Connect the LRESET pin to GND via a bypass capacitor (capacity: approx. 10 μF).**

Pin Identification

ANI10, ANI15:	Analog Input	P50 to P57:	Port 5
ANIDS00, ANIDS01,		P60, P61:	Port 6
ANIDS10, ANIDS11,		P81, P82:	Port 8
ANIDS20, ANIDS21,		P90 to P97:	Port 9
ANIDS30, ANIDS31:	ΔΣ Analog Input	P100:	Port 10
AV_{DD}:	Analog Power Supply	P120 to P122, P124:	Port 12
AV_{REF}:	Analog Reference Voltage	P140 to P147:	Port 14
AV_{REFIO}:	ΔΣ Analog Reference Voltage	P152, P157:	Port 15
CAPH, CAPL:	Capacitor for LCD	REGC:	Regulator Capacitance
CF:	Calibration Frequency Output	$\overline{\text{RESET}}$:	Reset
COM0 to COM3:	LCD Common Output	RTC1HZ:	Real-time Counter Correction
EXCLK:	External Clock Input		Clock (1Hz) Output
	(Main system clock)	RxD0, RxD2, RxD3:	Receive Data
EXCLKS0,		$\overline{\text{SCK20}}$:	Serial Clock Input/Output
EXCLKS1:	External Clock Input	SCL0, SCL20:	Serial Clock Input/Output
	(Subsystem Clock)	SDA0, SDA20 :	Serial Data Input/Output
EXLVI:	External Potential Input	SEG0 to SEG39:	LCD Segment Output
	for Low Voltage Detector	SI20:	Serial Data Input
FLMD0:	Flash Programming Mode	SO20:	Serial Data Output
FXTOUT:	Subsystem Clock Output	TI02, TI04, TI07:	Timer Input
IC0 to IC6:	Internal Connection	TO02, TO05, TO07:	Timer Output
INTP0, INTP3		TOOL0:	Data Input/Output for Tool
INTP6, INTP9:	External Interrupt Input	TOOL1:	Clock Output for Tool
LAV_{DD}:	ΔΣ Analog Power Supply	TxD0, TxD2, TxD3:	Transmit Data
LAV_{SS}	ΔΣ Analog Ground	V_{DD}:	Power Supply
LP01:	Port LP0	V_{LC0} to V_{LC2}:	LCD Power Supply
LRESET:	Reset	V_{SS}:	Ground
LV_{DD}:	Power Supply	X1, X2:	Crystal Oscillator
LV_{SS}:	Ground		(Main system clock)
P00, P01:	Port 0	XT1, XT2:	Crystal Oscillator (Subsystem
P10 to P12, P16:	Port 1		Clock)
P33:	Port 3	ZX1:	Zero Cross Detection Signal Output
P40, P41:	Port 4	32KHZCLK:	32 kHz Clock Output

1.4 Block Diagram



1.5 Outline of Functions

(1/2)

Item			78K0R/LG3-M μPD78F8070
Internal memory	Flash memory (self-programming supported)		128 KB
	RAM		7 KB
Memory space			1 MB
Main system clock (Oscillation frequency)	High-speed system clock		X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 2 to 20 MHz: V_{DD} = 2.7 to 3.6 V, 2 to 5 MHz V_{DD} = 1.8 to 2.7 V
	Internal high-speed oscillation clock		Internal oscillation 1 MHz (TYP.) or 8 MHz (TYP.) selected by an option byte
Subsystem clock (Oscillation frequency)			XT1 (crystal) oscillation, external subsystem clock input (EXCLKS1) 32.768 kHz (TYP.)
Internal low-speed oscillation clock (For WDT)			Internal oscillation 30 kHz (TYP.)
General-purpose register			8 bits × 32 registers (8 bits × 8 registers × 4 banks)
Minimum instruction execution time			0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)
			0.125 μs (Internal high-speed oscillation clock: fIH = 8 MHz (TYP.) operation)
			30.5 μs (Subsystem clock: f_{SUB} = 32.768 kHz operation)
Instruction set			• 8-bit operation, 16-bit operation • Multiply (16 bits × 16 bits) • Bit manipulation (Set, reset, test, and Boolean operation), etc.
I/O port	Total		45
	I/O	CMOS	39
		N-ch O.D.	2
	Output	CMOS	1
	Input	CMOS	3
Timer			• 16-bit timer: 12 channels • Watchdog timer: 1 channel • Real-time counter: 1 channel • Real-time counter 2: 1 channel
	Timer outputs		3 (PWM output: 3 (Timer array unit 0))
	RTC outputs		1 • 1 Hz (Subsystem clock: f_{SUB} = 32.768 kHz) • 32.768 kHz (Subsystem clock: f_{SUB} = 32.768 kHz)
10-bit successive approximation type A/D converter			2 channels
24-bit $\Delta\Sigma$-type A/D converter			4 channels

(2/2)

Item		78K0R/LG3-M
		μPD78F8070
Serial interface	UART supporting LIN-bus	1 channel
	CSI/UART/ simplified I^2C	1 channel
	UART	1 channel
	Multimaster I^2C	1 channel
LCD controller/driver		Internal voltage boosting method, capacitor split method, and external resistance division method are switchable.
	Segment signal output	40
	Common signal output	4
Multiplier/divider		16 bits × 16 bits = 32 bits (multiplication), 32 bits ÷ 32 bits = 32 bits, 32-bit remainder (division)
DMA controller		2 channels
Vectored interrupt sources	Internal	32
	External	4
Power calculation circuit		Provided
Power quality measurement circuit		Provided
Digital frequency conversion circuit		Provided
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector • Internal reset by illegal instruction execution[Note]
Power-on-clear circuit		• Power-on-reset: 1.61 ±0.09 V • Power-down-reset: 1.59 ±0.09 V
Low-voltage detector	VDD voltage detector	1.91 V to 3.45 V (11stages)
	EXLVI voltage detector	1.21 V
On-chip debug function		Provided
BCD adjustment		
Power supply voltage		VDD = 1.8 to 3.6 V
Operating ambient temperature		TA = −40 to +85°C
Package		100-pin plastic LQFP (fine pitch) (14 × 14)

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

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

There are four types of pin I/O buffer power supplies: AV_{DD}, LAV_{DD}, LV_{DD}, and V_{DD}. The relationship between these power supplies and the pins is shown below.

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

Power Supply	Corresponding Pins
AV_{DD}	P152, P157
LAV_{DD}	ANIDS00, ANIDS01, ANIDS10, ANIDS11, ANIDS20, ANIDS21, ANIDS30, ANIDS31 pins
LV_{DD}	CF, EXCLKS1, LP01, XT1, XT2, ZX1, LRESET, FXTOUT, TO05, P16 pins
V_{DD}	Pins other than port

(1) Port functions (1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	CAPH
P01				CAPL
P10	I/O	Port 1. 3-bit I/O port and 1-bit output port Input/output can be specified in 1-bit units. Input of P10 and P11 can be set to TTL buffer. Output of P10 to P12 can be set to N-ch open-drain output (V_{DD} tolerance). For input of P10 to P12, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{SCK20}$/SCL20
P11				SI20/RxD2/SDA20/ INTP6
P12				SO20/TxD2/TO02
<R> P16	Output		Output port Note 1	TO05
P33	I/O	Port 3. 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI07/TO07/INTP3
P40 Note 2	I/O	Port 4. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOOL0
P41				TOOL1
P50	I/O	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.	Input port	SEG39/RxD3
P51				SEG38/TxD3
P52				SEG37/TI02
P53				SEG36/TI04
P54 to P57				SEG35 to SEG32
P60	I/O	Port 6. 2-bit I/O port. Output is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCL0
P61				SDA0
P81	I/O	Port 8. 2-bit I/O port. Inputs/output can be specified in 1-bit units. Output of P82 can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	RxD0/ INTP9
P82				TxD0
P90 to P97	I/O	Port 9. 8-bit I/O port. Inputs/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG31 to SEG24
P100	I/O	Port 10. 1-bit I/O port. Inputs/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG15

<R> **Notes 1.** When using P16/TO05, after reset release, be sure to clear PM16 bit of PM1 and PM20 bit of PM2 to 0, and to set P20 bit of P2 to 1.

2. If on-chip debugging is enabled by using an option byte, be sure to pull up the P40/TOOL0 pin externally.

(1) Port functions (2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P120	I/O	Port 12. 1-bit I/O port and 3-bit input port. For only P120, input/output can be specified in 1-bit units. For only P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121	Input			X1
P122				X2/EXCLK
P124				EXCLKS0
P140 to P147	I/O	Port 14. 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.	Input port	SEG23 to SEG16
P152	I/O	Port 15. 2-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI10
P157				ANI15
LP01	I/O	Port LP0. 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	ZX1

(2) Non-port functions (1/3)

Function Name	I/O	Function	After Reset	Alternate Function
ANI10	Input	10-bit successive approximation type A/D converter analog input	Digital input port	P152
ANI15				P157
AV_{REF}	Input	10-bit successive approximation type A/D converter reference voltage input	Input	–
ANIDS00	Input	24-bit ΔΣ-type A/D converter analog input. ANIDSx0 is a negative input pin, and ANIDSx1 is a positive input pin. Note that the channel (voltage or current) that corresponds to each input pin varies according to whether the two-wire mode or three-wire mode is in use (See **CHAPTER 11 24-BIT ΔΣ-TYPE A/D CONVERTER**).	Input	–
ANIDS01				–
ANIDS10				–
ANIDS11				–
ANIDS20				–
ANIDS21				–
ANIDS30				–
ANIDS31				–
LAV_{DD}	–	Positive power supply for 24-bit ΔΣ-type A/D converter	–	–
<R> AV_{REFIO}	I/O	24-bit ΔΣ-type A/D converter reference voltage	Output port	–
LAV_{SS}	–	24-bit ΔΣ-type A/D converter ground potential. Make the same potential as V_{SS}.	–	–
SEG0 to SEG14	Output	LCD controller/driver segment signal outputs	Output	–
SEG15			Input port	P100
SEG16 to SEG23				P147 to P140
SEG24 to SEG31				P97 to P90
SEG32 to SEG39				P57 to P50
CF	Output	Calibration frequency output	Output	–
COM0 to COM3	Output	LCD controller/driver common signal outputs	Output	–
V_{LC0} to V_{LC2}	–	LCD drive voltage	–	–
CAPH	–	Connecting a capacitor for LCD controller/driver	Input port	P00
CAPL				P01
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P120/EXLVI
INTP3				P33/TI07/TO07
INTP6				P11/SI20/RxD2/SDA20
INTP9				P81/RxD0
LV_{DD}	–	Real-time counter, power calculation circuit, subsystem clock oscillator, and positive power supply for port LP	–	–
<R> LV_{SS}	–	Real-time counter, power calculation circuit, subsystem clock oscillator, and ground potential for port LP. Short to the V_{SS} pin using wiring that is as short as possible and as thick as possible.	–	–
REGC	–	Connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF).	–	–
RTC1HZ	Output	Real-time counter correction clock (1 Hz) output	Output	32KHZCLK
<R> $\overline{RESET}$	Input	System reset input. Connect to GND via a bypass capacitor (capacity: approx. 47 μF).	–	–

(2) Non-port functions (2/3)

Function Name	I/O	Function	After Reset	Alternate Function
<R> LRESET	Input	System reset input. However, this pin must be connected directly to LV_{DD} or via resistance. Also connect to GND via a bypass capacitor (capacity: approx. 10 μF).	–	–
RxD0	Input	Serial data input to UART0	Input port	P81/INTP9
RxD2		Serial data input to UART2		P11/SI20/SDA20/INTP6
RxD3		Serial data input to UART3		P50/SEG39
$\overline{SCK20}$	I/O	Clock input/output for CSI20	Input port	P10/SCL20
SCL0	I/O	Clock input/output for I^2C	Input port	P60
SCL20	I/O	Clock input/output for simplified I^2C	Input port	P10/$\overline{SCK20}$
SDA0	I/O	Serial data I/O for I^2C	Input port	P61
SDA20	Input	Serial data I/O for simplified I^2C		P11/SI20/RxD2/INTP6
SI20		Serial data input to CSI20	Input port	P11/RxD2/SDA20/INTP6
SO20	Output	Serial data output from CSI20	Input port	P12/TxD2/TO02
TI02	Input	External count clock input to 16-bit timer 02	Input port	P52/SEG37
TI04		External count clock input to 16-bit timer 04		P53/SEG36
TI07		External count clock input to 16-bit timer 07		P33/TO07/INTP3
TO02		16-bit timer 02 output		P12/SO20/TxD2
TO05		16-bit timer 05 output		P16
TO07		16-bit timer 07 output		P33/TI07/INTP3
TxD0	Output	Serial data output from UART0	Input port	P82
TxD2		Serial data output from UART2		P12/SO20/TO02
TxD3		Serial data output from UART3		P51/SEG38
X1	–	Resonator connection for main system clock	Input port	P121
X2	–		Input port	P122/EXCLK
EXCLK	Input	External clock input for main system clock	Input port	P122/X2
EXCLKS0	Input	External clock input for subsystem clock	Input port	P124
EXCLKS1	Input	External clock input for subsystem clock However, be sure to connect the output from FXTOUT pin to the input of the P124/EXCLKS0 pin on the used board.	Input	XT1
FXTOUT	Output	Subsystem clock output However, be sure to connect the output from FXTOUT pin to the input to the P124/EXCLKS0 pin on the used board.	Output	–
XT1	–	Resonator connection for subsystem clock	–	EXCLKS1
XT2	–		–	–
<R> ZX1	Output	Zero-crossing detection signal output	Input port	LP01
32KHZCLK	Output	Real-time counter clock (32 kHz) output	Output	RTC1HZ
V_{DD}	–	Positive power supply	–	–
AV_{DD}	–	Positive power supply for P152, P157 and 10-bit successive approximation type A/D converter	–	–

(2) Non-port functions (3/3)

Function Name	I/O	Function	After Reset	Alternate Function
<R> V_{SS}	–	Ground potential. Short to the LV_{SS} pin using wiring that is as short as possible and as thick as possible.	–	–
FLMD0	–	Flash memory programming mode setting	–	–
TOOL0	I/O	Data I/O for flash memory programmer/debugger	Input port	P40
TOOL1	Output	Clock output for debugger	Input port	P41
IC0 to IC6	–	Internal connection pin IC0 to IC2, IC4 to IC6 must be opened. IC3 pin must be connected to LV_{SS}.	–	–

2.2 Description of Pin Functions

2.2.1 P00, P01

They function as an I/O port. They can also be used for connecting a capacitor for LCD controller/driver.

78K0R/LG3-M (100 pin: μPD78F8070)
P00/CAPH
P01/CAPL

The following operation modes can be specified in 1-bit units.

(1) Port mode

P00 and P01 function as an I/O port. They can be set to input or output port in 1-bit units using port mode register 0 (PM0). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

(2) Control mode

P00 and P01 function as connecting a capacitor for LCD controller/driver.

(a) CAPH, CAPL

These are the pins for connecting a capacitor for LCD controller/driver.

Caution To use P00/CAPH, P01/CAPL as a general-purpose port, set bit 5 (MDSET1) and bit 4 (MDSET0) of LCD mode register (LCDMD) to "0", which is the same as their default status setting.

2.2.2 P10 to P12, P16

P10 to P12, P16 function as an I/O port. This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, and timer output.

Input to the P10 and P11 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 P10 to P12 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 1 (POM1).

78K0R/LG3-M (100 pin: μPD78F8070)
P10/$\overline{SCK20}$/SCL20
P11/SI20/RxD2/SDA20/INTP6
P12/SO20/TxD2/TO02
P16/TO05

The following operation modes can be specified in 1-bit units.

(1) Port mode

P10 to P12, P16 function as an I/O port. They can be set to input or output port in 1-bit units using port mode register 1 (PM1). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

(2) Control mode

P10 to P12, P16 function as serial interface clock I/O, data I/O, timer output, and external interrupt request input.

(a) $\overline{SCK20}$

This is serial clock I/O pin of serial interface CSI20.

(b) SI20

This is serial data input pin of serial interface CSI20.

(c) SO20

This is serial data output pin of serial interface CSI20.

(d) SCL20

This is serial clock I/O pin of serial interface IIC20 (simplified I^2C).

(e) SDA20

This is serial data I/O pin of serial interface IIC20 (simplified I^2C).

(f) RxD2

This is serial data input pin of serial interface UART2.

(g) TxD2

This is serial data output pin of serial interface UART2.

(h) TO02

This is the timer output pin of 16-bit timers 02.

(i) INTP6

This is the external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

Cautions 1. To use P10/$\overline{\text{SCK20}}$/SCL20 and P11/SI20/RxD2/SDA20/INTP6 as a general-purpose port, note the serial array unit 1 setting. For details, refer to Table 12-7 Relationship Between Register Settings and Pins (Channel 0 of unit 1: CSI20, UART2 Reception, IIC20).

2. To use P12/TO02/SO20/TxD2 as a general-purpose port, set bit 2 (TO02) of timer output register 0 (TO0) and bit 2 (TOE02) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting. And as a general-purpose port, note the serial array unit 1 setting. For details of serial array unit 1 setting, refer to Table 12-7 Relationship Between Register Settings and Pins (Channel 0 of unit 1: CSI20, UART2 Reception, IIC20).

3. To use P16/TO05 as a general-purpose port, set bit 5 (TO05) of timer output register 0 (TO0) and bit 5 (TOE05) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting. Also, be sure to clear 0 to the PM20 bit of PM2 after reset is released, and set 1 to the P20 bit of P2 after reset is released.

2.2.3 P33

P33 functions as an I/O port. This port can also be used for timer I/O and external interrupt request input.

78K0R/LG3-M (100pin: μPD78F8070)
P33/TI07/TO07/INTP3

The following operation modes can be specified in 1-bit units.

(1) Port mode

P33 functions as an I/O port. It can be set to input or output port in 1-bit units using port mode register 3 (PM3). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

(2) Control mode

P33 functions as timer I/O and external interrupt request input.

(a) TI07

This is the timer input pin of 16-bit timer 07.

(b) TO07

This is the timer output pin of 16-bit timer 07.

(c) INTP3

This is the external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

Caution To use P33/TO07/TI07/INTP3 as a general-purpose port, set bit 7 (TO07) of timer output register 0 (TO0), and bit 7 (TOE07) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

2.2.4 P40, P41

P40 and P41 function as an I/O port. These pins also function as data I/O for a flash memory programmer/debugger and clock output for a debugger.

78K0R/LG3-M (100 pin: μPD78F8070)
P40/TOOL0
P41/TOOL1

The following operation modes can be specified in 1-bit units.

(1) Port mode

P40 and P41 function as an I/O port. P40 and P41 can be set to input or output port in 1-bit units using port mode register 4 (PM4). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 4 (PU4).

Be sure to connect an external pull-up resistor to P40 when on-chip debugging is enabled (by using an option byte).

(2) Control mode

P40 and P41 function as data I/O for a flash memory programmer/debugger and clock output for a debugger.

(a) TOOL0

This is a data I/O pin for a flash memory programmer/debugger.

Be sure to pull up this pin externally when on-chip debugging is enabled (pulling it down is prohibited).

(b) TOOL1

This is a clock output pin for a debugger.

When the on-chip debug function is used, P41/TOOL1 pin can be used as follows by the mode setting on the debugger.

1-line mode: can be used as a port (P41).

2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).

Caution The function of the P40/TOOL0 pin varies as described in (a) to (c) below.
In the case of (b) or (c), make the specified connection.

(a) In normal operation mode and when on-chip debugging is disabled (OCDENSET = 0) by an option byte (000C3H)
=> Use this pin as a port pin (P40).

(b) In normal operation mode and when on-chip debugging is enabled (OCDENSET = 1) by an option byte (000C3H)
=> Connect this pin to V_{DD} via an external resistor, and always input a high level to the pin before reset release.

(c) When on-chip debug function is used, or in write mode of flash memory programmer
=> Use this pin as TOOL0.
Directly connect this pin to the on-chip debug emulator or a flash memory programmer, or pull it up by connecting it to V_{DD} via an external resistor.

2.2.5 P50 to P57

P50 to P57 function as an I/O port. This port can also be used for serial interface data I/O, timer input, and segment output of LCD controller/driver.

78K0R/LG3-M (100 pin: μPD78F8070)
P50/RxD3/SEG39
P51/TxD3/SEG38
P52/TI02/SEG37
P53/TI04/SEG36
P54/SEG35
P55/SEG34
P56/SEG33
P57/SEG32

The following operation modes can be specified in 1-bit units.

(1) Port mode

P50 to P57 function as an I/O port. P50 to P57 can be set to input or output port in 1-bit units using port mode register 5 (PM5). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 5 (PU5).

(2) Control mode

P50 to P57 function as serial interface data I/O, timer input, and segment output of LCD controller/driver.

(a) RxD3

This is a serial data input pin of serial interface UART3.

(b) TxD3

This is a serial data output pin of serial interface UART3.

(c) TI02, TI04

These are the timer input pins of 16-bit timers 02 and 04.

(d) SEG32 to SEG39

These are the segment output pins of LCD controller/driver.

2.2.6 P60, P61

P60 and P61 function as an I/O port. This port can also be used for s serial interface IICA data I/O and clock I/O.

78K0R/LG3-M (100 pin: μPD78F8070)
P60/SCL0
P61/SDA0

The following operation modes can be specified in 1-bit units.

(1) Port mode

P60 and P61 function as an I/O port. P60 and P61 can be set to input port or output port in 1-bit units using port mode register 6 (PM6).

Output of P60 and P61 is N-ch open-drain output (6 V tolerance).

(2) Control mode

P60 and P61 function as serial interface IICA clock I/O and data I/O.

(a) SCL0

This is a serial clock I/O pin of serial interface IICA.

(b) SDA0

This is a serial data I/O pin of serial interface IICA.

Caution When using P60/SCL0 and P61/SDA0 as a general-purpose port, stop the operation of serial interface IICA.

2.2.7 P81, P82

P81 and P82 function as an I/O port. This port can also be used for serial interface data I/O, and external interrupt request input.

Output from the P82 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 8 (POM8).

78K0R/LG3-M (100 pin: μPD78F8070)
P81/RxD0/INTP9
P82/TxD0

The following operation modes can be specified in 1-bit units.

(1) Port mode

P81 and P82 function as an I/O port. they can be set to input or output port in 1-bit units using port mode register 8 (PM8). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 8 (PU8).

(2) Control mode

P81 and P82 function as serial interface data I/O and external interrupt request input.

(a) RxD0

This is a serial data input pin for serial interface UART0.

(b) TxD0

This is a serial data output pin for serial interface UART0.

(c) INTP9

This is the external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

Caution To use P81/RxD0/INTP9, and P82/TxD0, as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 12-5 Relationship Between Register Settings and Pins (Channel 0 of unit 0: UART0 Reception).

2.2.8 P90 to P97

P90 to P97 function as an I/O port. This port can also be used for segment output of LCD controller/driver.

78K0R/LG3-M (100 pin: μPD78F8070)
P90/SEG31
P91/SEG30
P92/SEG29
P93/SEG28
P94/SEG27
P95/SEG26
P96/SEG25
P97/SEG24

The following operation modes can be specified in 1-bit units.

(1) Port mode

P90 to P97 function as an I/O port. P90 to P97 can be set to input or output port in 1-bit units using port mode register 9 (PM9). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 9 (PU9).

(2) Control mode

P90 to P97 function as segment output of LCD controller/driver (SEG24 to SEG31).

(a) SEG24 to SEG31

These are the segment output pins for the LCD controller/driver.

2.2.9 P100

P100 functions as an I/O port. This port can also be used for segment output of LCD controller/driver.

78K0R/LG3-M (100 pin: μPD78F8070)
P100/SEG15

The following operation modes can be specified in 1-bit units.

(1) Port mode

P100 functions as an I/O port. It can be set to input or output port in 1-bit units using port mode register 10 (PM10). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 10 (PU10).

(2) Control mode

P100 functions as segment output of LCD controller/driver (SEG15).

(a) SEG15

This is the segment output pin of the LCD controller/driver.

2.2.10 P120 to P122, P124

P120 functions as an I/O port. P121, P122, and P124 function as an input port. These pins also function as potential input for external low-voltage detection, connecting resonator for main system clock, external clock input for main system clock, external clock input for subsystem clock, and external interrupt request input.

78K0R/LG3-M (100 pin: μPD78F8070)
P120/INTP0/EXLVI
P121/X1
P122/X2/EXCLK
P124/EXCLKS0

The following operation modes can be specified in 1-bit units.

(1) Port mode

P120 functions as an I/O port. P120 can be set to input port or output port using port mode register 12 (PM12). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

P121, P122 and P124 function as an input port.

(2) Control mode

P120 to P122, and P124 function as potential input for external low-voltage detection, connecting resonator for main system clock, external clock input for main system clock, external clock input for subsystem clock, and external interrupt request input.

(a) EXLVI

This is a potential input pin for external low-voltage detection.

(b) X1, X2

These are the pins for connecting a resonator for main system clock.

(c) EXCLK

This is an external clock input pin for main system clock.

(d) EXCLKS0

This is an external clock input for subsystem clock.

Be sure to connect the output from FXTOUT pin to the input of the P124/EXCLKS0 pin on the used board.

(e) INTP0

This is an external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

Cautions 1. The function setting on P121, P122 and P124 are available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.

2. Be sure to use P124 as an external clock input for subsystem clock (EXCLKS0). When EXCLKS0 pin is used, the setting must be done by Clock Operation Mode Control Register (CMC) and Port Function Control Register (PORTCTL).

2.2.11 P140 to P147

P140 to P147 function as an I/O port. This port can also be used for segment output of LCD controller/driver.

78K0R/LG3-M (100 pin: μPD78F8070)
P140/SEG23
P141/SEG22
P142/SEG21
P143/SEG20
P144/SEG19
P145/SEG18
P146/SEG17
P147/SEG16

The following operation modes can be specified in 1-bit units.

(1) Port mode

P140 to P147 function as an I/O port. P140 to P147 can be set to input or output port in 1-bit units using port mode register 14 (PM14). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 14 (PU14).

(2) Control mode

P140 to P147 function as segment output of LCD controller/driver (SEG16 to SEG23).

(a) SEG16 to SEG23

These are the segment output pins for the LCD controller/driver.

2.2.12 P152, P157

P152 and P157 function as an I/O port. This port can also be used for 10-bit successive approximation type A/D converter analog input.

78K0R/LG3-M (100 pin: μPD78F8070)
P152/ANI10
P157/ANI15

The following operation modes can be specified in 1-bit units.

(1) Port mode

P152 and P157 function as an I/O port. P152 and P157 can be set to input or output port in 1-bit units using port mode register 15 (PM15).

(2) Control mode

P152 and P157 function as 10-bit successive approximation type A/D converter analog input.

(a) ANI10, ANI15

These are 10-bit successive approximation type A/D converter analog input pins.

Cautions 1. P152/ANI10 and P157/ANI15 are set in the digital input (general-purpose port) mode after release of reset.

2. When using at least one port of P152/ANI10 and P157/ANI15 as a digital port, set AV_{DD} to the same potential as V_{DD}.

2.2.13 LP01

LP01 functions as an I/O port. It also functions as pin for zero-crossing detection signal output of the power quality measurement circuit.

78K0R/LG3-M (100 pin: μPD78F8070)
LP01/ZX1

The following operation modes can be specified in 1-bit units.

(1) Port mode

It functions as an I/O port. It can be set to input or output port in 1-bit units using port mode register LP0 (LPM0). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register LP0 (LPU0).

(2) Control mode

LP01 functions as zero-crossing detection signal output of the power quality measurement circuit.

(a) ZX1

This is pin for zero-crossing detection signal output.

2.2.14 ANIDS00, ANIDS01, ANIDS10, ANIDS11, ANIDS20, ANIDS21, ANIDS30, ANIDS31

These pins are the 24-bit ΔΣ-type A/D converter analog input.

ANIDSx0 is a negative input pin, and ANIDSx1 is a positive input pin.

Note that the channel (voltage or current) that corresponds to each input pin varies according to whether the two-wire mode or three-wire mode is in use (See **CHAPTER 11 24-BIT ΔΣ-TYPE A/D CONVERTER**).

2.2.15 CF

This is the calibration frequency output pin of digital frequency conversion circuit.

2.2.16 COM0 to COM3

These are common outputs of LCD controller/driver.

2.2.17 SEG0 to SEG14

These are segment outputs of LCD controller/driver.

2.2.18 EXCLKS1

This is an external clock input pin for subsystem clock.

2.2.19 FXTOUT

This is a subsystem clock output pin.

Be sure to connect the output from FXTOUT pin to the input of the P124/EXCLKS0 pin on the used board.

2.2.20 IC0 to IC6

These are an internal connection pin.

IC0 to IC2, IC4 to IC6 must be opened.

IC3 pin must be connected to LV_{SS}.

2.2.21 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

When the external reset pin is not used, connect this pin directly to V_{DD} or via a resistor.

When the external reset pin is used, design the circuit based on V_{DD}.

<R> This pin must also be connected to GND via a bypass capacitor (capacity: approx. 47 μF).

2.2.22 LRESET

This is a system reset input pin of low level active.

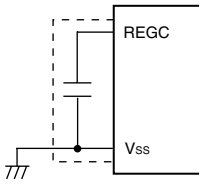
LRESET pin must be connected directly to LV_{DD} or via resistance.

<R> This pin must also be connected to GND via a bypass capacitor (capacity: approx. 10 μF).

2.2.23 REGC

This is the pin for connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF).

Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

2.2.24 RTC1HZ

This is the real-time counter correction clock (1 Hz) output pin.

2.2.25 XT1, XT2

These are the pins for connecting a resonator for subsystem clock.

2.2.26 32KHZCLK

This is the real-time counter clock (32 kHz) output pin.

2.2.27 FLMD0

This is a pin for setting flash memory programming mode.

Perform either of the following processing.

(a) In normal operation mode

It is recommended to leave this pin open during normal operation.

The FLMD0 pin must always be kept at the V_{SS} level before reset release but does not have to be pulled down externally because it is internally pulled down by reset. However, pulling it down must be kept selected (i.e., FLMDPUP = "0", default value) by using bit 7 (FLMDPUP) of the background event control register (BECTL) (see **28.5 (1) Back ground event control register (BECTL)**). To pull it down externally, use a resistor of 200 kΩ or smaller.

Self programming and the rewriting of flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V_{SS} pin.

(b) In self programming mode

It is recommended to leave this pin open when using the self programming function. To pull it down externally, use a resistor of 100 kΩ to 200 kΩ.

In the self programming mode, the setting is switched to pull up in the self programming library.

(c) In flash memory programming mode

Directly connect this pin to a flash memory programmer when data is written by the flash memory programmer. This supplies a writing voltage of the V_{DD} level to the FLMD0 pin.

The FLMD0 pin does not have to be pulled down externally because it is internally pulled down by reset. To pull it down externally, use a resistor of 1 kΩ to 200 kΩ.

2.2.28 AV_{DD}, AV_{REF}, LAVDD, AV_{REFIO}, LAV_{SS}, LV_{DD}, LV_{SS}, V_{DD}, V_{LC0} to V_{LC2}, V_{SS}

(1) AV_{DD}

This is the positive power supply pin of 10-bit successive approximation type A/D converter and P152 and P157.

When using at least one port of port 15 as a digital port, or when not using the 10-bit successive approximation type A/D converter, set AV_{DD} to the same potential as V_{DD}.

(2) AV_{REF}

This is the reference voltage input pin of 10-bit successive approximation type A/D converter.

When not using the 10-bit successive approximation type A/D converter, set AV_{REF} to the same potential as AV_{DD} or V_{DD}.

(3) LAV_{DD}

This is the positive power supply of 24-bit ΔΣ-type A/D converter.

(4) AV_{REFIO}

This is the reference voltage I/O pin of 24-bit ΔΣ-type A/D converter.

(5) LAV_{SS}

This is the 24-bit ΔΣ-type A/D converter ground potential pin. When not using the 24-bit ΔΣ-type A/D converter, directly connect this pin to LV_{SS}.

(6) LV_{DD}

This is the positive power supply pin of real-time counter, power calculation circuit, subsystem clock oscillator, and port LP.

(7) LV_{SS}

This is the ground potential pin of real-time counter, power calculation circuit, subsystem clock oscillator, and port LP. When not using real-time counter, power calculation circuit, subsystem clock oscillator, and port LP, directly connect this pin to V_{SS}.

<R> Short this pin to the V_{SS} pin using wiring that is as short as possible and as thick as possible.

(8) V_{DD}

This is the positive power supply pin.

(9) V_{LC0} to V_{LC2}

These pins are the power supply voltage pins for driving the LCD.

(10) V_{SS}

This is the ground potential pin.

<R> Short this pin to the LV_{SS} pin using wiring that is as short as possible and as thick as possible.

2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

Table 2-2 shows the types of pin I/O circuits and the recommended connections of unused pins.

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/CAPH	12-H	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P01/CAPL			
P10/$\overline{SCK20}$/SCL20	5-AN		
P11/SI20/RxD2/SDA20/INTP6			
P12/SO20/TxD2/TO02	5-AG		
P16/TO05	3-C	Output	
P33/TI07/TO07/INTP3	8-R	I/O	
P40/TOOL0		I/O	<When on-chip debugging is enabled> Pull this pin up (pulling it down is prohibited). <When on-chip debugging is disabled> Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P41/TOOL1	5-AG		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P50/SEG39/RxD3	17-Z		<When setting to port I/O> Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P51/SEG38/TxD3	17-Y		
P52/SEG37/TI02	17-Z		
P53/SEG36/TI04			
P54/SEG35 to P57/SEG32	17-Y		
P60/SCL0	13-R		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P61/SDA0			
P81/RxD0/INTP9	8-R		
P82/TxD0	5-AG		
P90/SEG31 to P97/SEG24	17-Y		<When setting to port I/O> Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P100/SEG15			
P120/INTP0/EXLVI	8-R		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121/X1[Note]	37-C	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK[Note]			

Note Use recommended connection above in input port mode (see **Figure 5-2 Format of Clock Operation Mode Control Register (CMC)**) when these pins are not used.

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P124/EXCLKS0	2-Y	Input	<When setting to port I/O> Independently connect to V_{DD} or V_{SS} via a resistor. <When setting to EXCLKS0> Be sure to connect the output from FXTOUT pin to the input of the P124/EXCLKS0 pin on the used board.
P140/SEG23 to P147/SEG16	17-Y	I/O	<When setting to port I/O> Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P152/ANI10[Note]	11-G		Input: Independently connect to AV_{DD} or V_{SS} via a resistor. Output: Leave open.
P157/ANI15[Note]			
SEG0 to SEG14	17-AC	Output	Leave open.
COM0 to COM3	18-G		
V_{LC0} to V_{LC2}	–	–	
AV_{DD}	–	–	<When setting one or more of P152 and P157 as digital port> Make this pin the same potential as V_{DD}. <When setting P152 and P157 as analog input> Make this pin to have a potential where 1.8 V ≤ AV_{DD} ≤ V_{DD}.
AV_{REF}	–	–	Set AV_{REF} to the same potential as AV_{DD} or V_{DD}.
$\overline{\text{RESET}}$	2	Input	Directly connect to V_{DD} or via a resistor. Connect to GND via a bypass capacitor (capacity: approx. 47μF).
REGC	–	–	Connect to V_{SS} via a capacitor (0.47 to 1μF).
FLMD0	2-W	–	Leave open or connect to V_{SS} via a resistor of 100 kΩ or more.
XT1/EXCLKS1	37-G	Input	Independently connect to LV_{DD} or LV_{SS} via a resistor.
XT2			Leave open.
LP01/ZX1	5-AG	I/O	Input: Independently connect to LV_{DD} or LV_{SS} via a resistor. Output: Leave open.
LRESET	2-Z	Input	<When setting to LRESET> Directly connect to LV_{DD} or via a resistor. Connect to GND via a bypass capacitor (capacity: approx. 10μF).
IC3	2	–	Directly connect to LV_{SS}.
CF	3-C	Output	Leave open.
RTC1HZ/32KHZCLK			
FXTOUT			<When setting to FXTOUT > Be sure to connect the output from FXTOUT pin to the input of the P124/EXCLKS0 pin on the used board.
LV_{SS}	–	–	Directly connect to V_{SS}. Connect to the V_{SS} pin using wiring that is as short as possible and as thick as possible.
LAV_{SS}	–	–	Directly connect to LV_{SS}.

Note P152/ANI10 and P157/ANI15 are set in the digital input port mode after release of reset.

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
ANIDS00, ANIDS01 to ANIDS30, ANIDS31	35	Input	Independently connect to LAV_{DD} or LAV_{SS} via a resistor.
AV_{REFIO}		I/O	Input: Independently connect to LAV_{SS} via a resistor. Output: Leave open.
IC0 to IC2, IC4 to IC6	–	–	Leave open.

Figure 2-1. Pin I/O Circuit List (1/4)

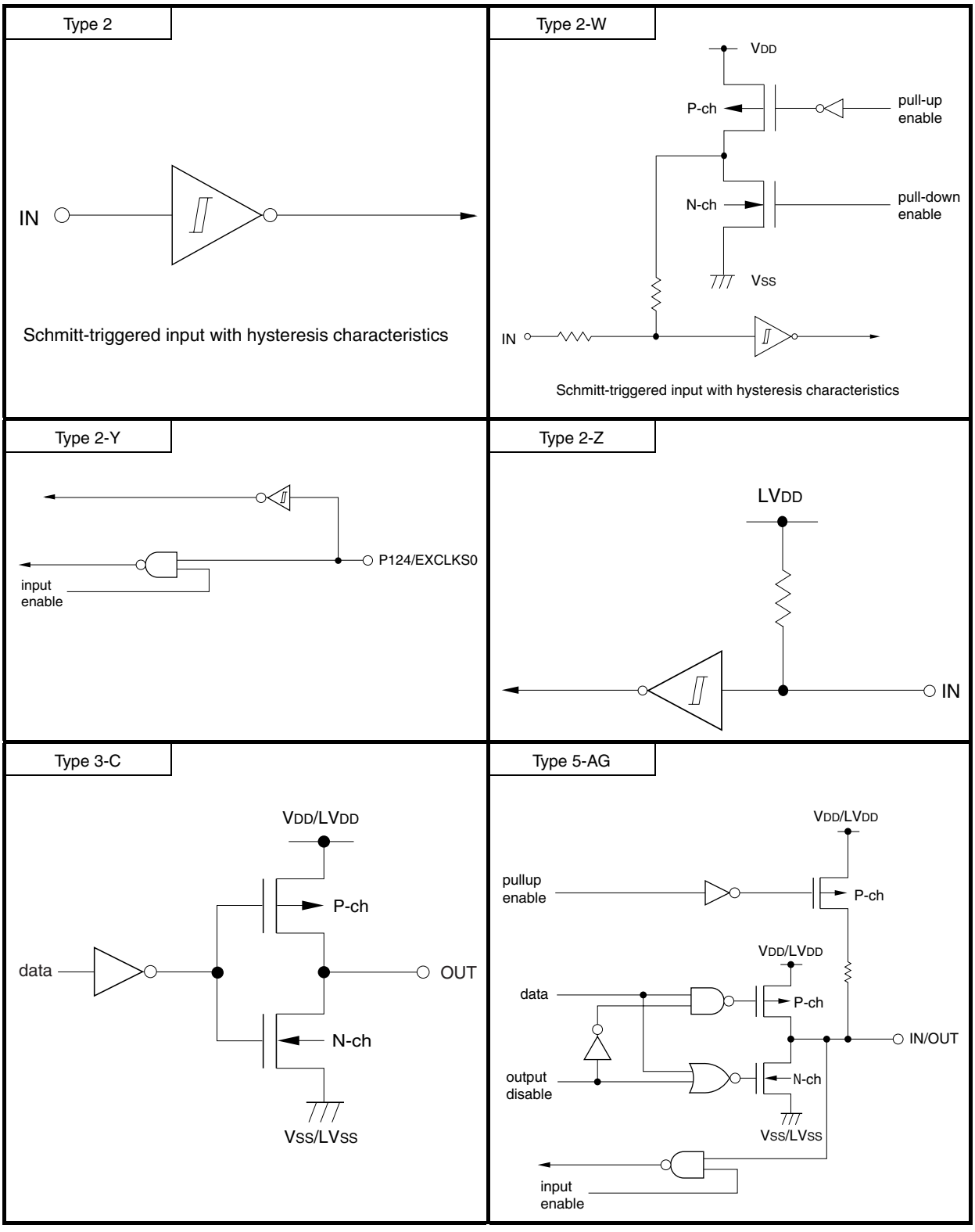


Figure 2-1. Pin I/O Circuit List (2/4)

<R>

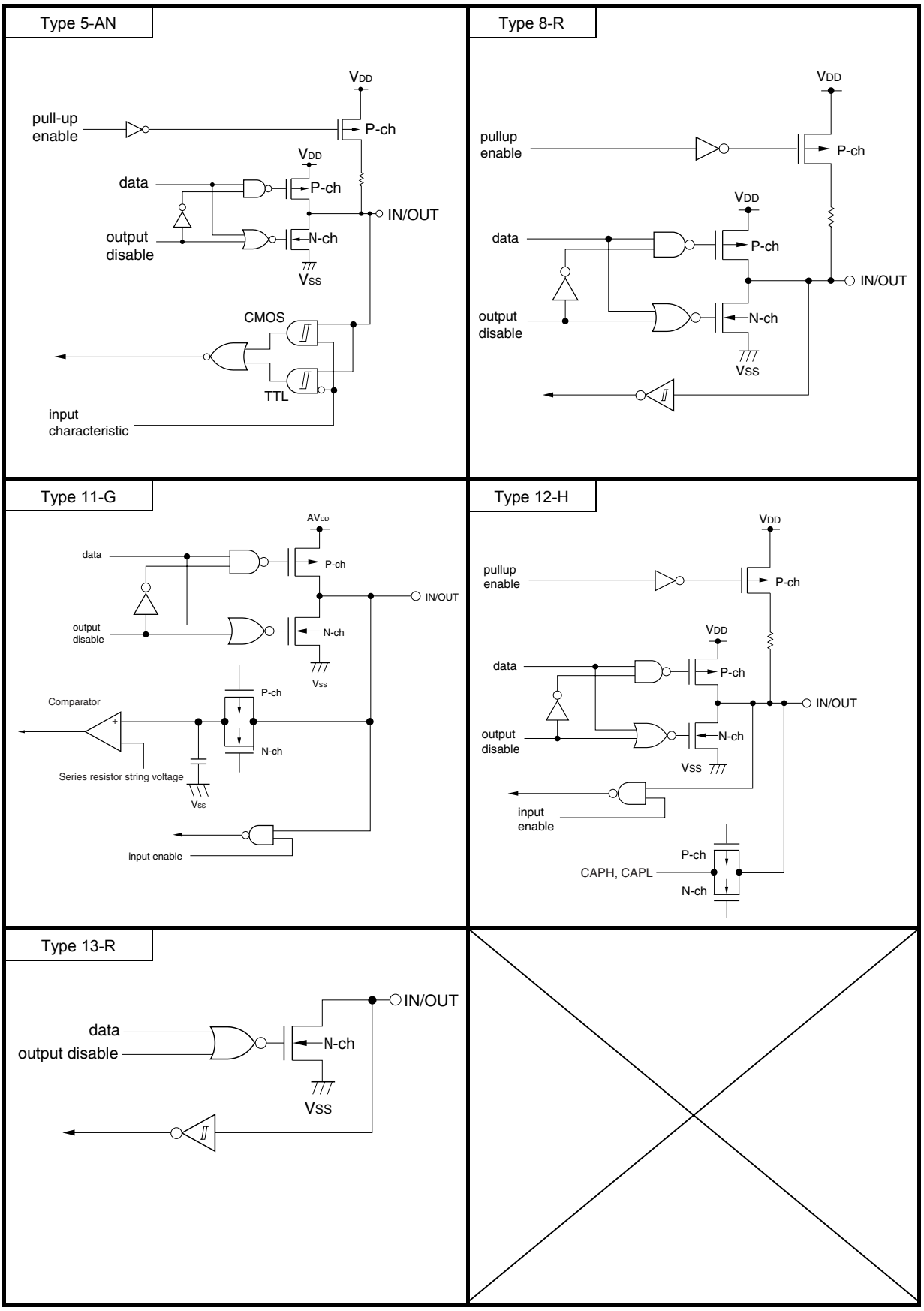


Figure 2-1. Pin I/O Circuit List (3/4)

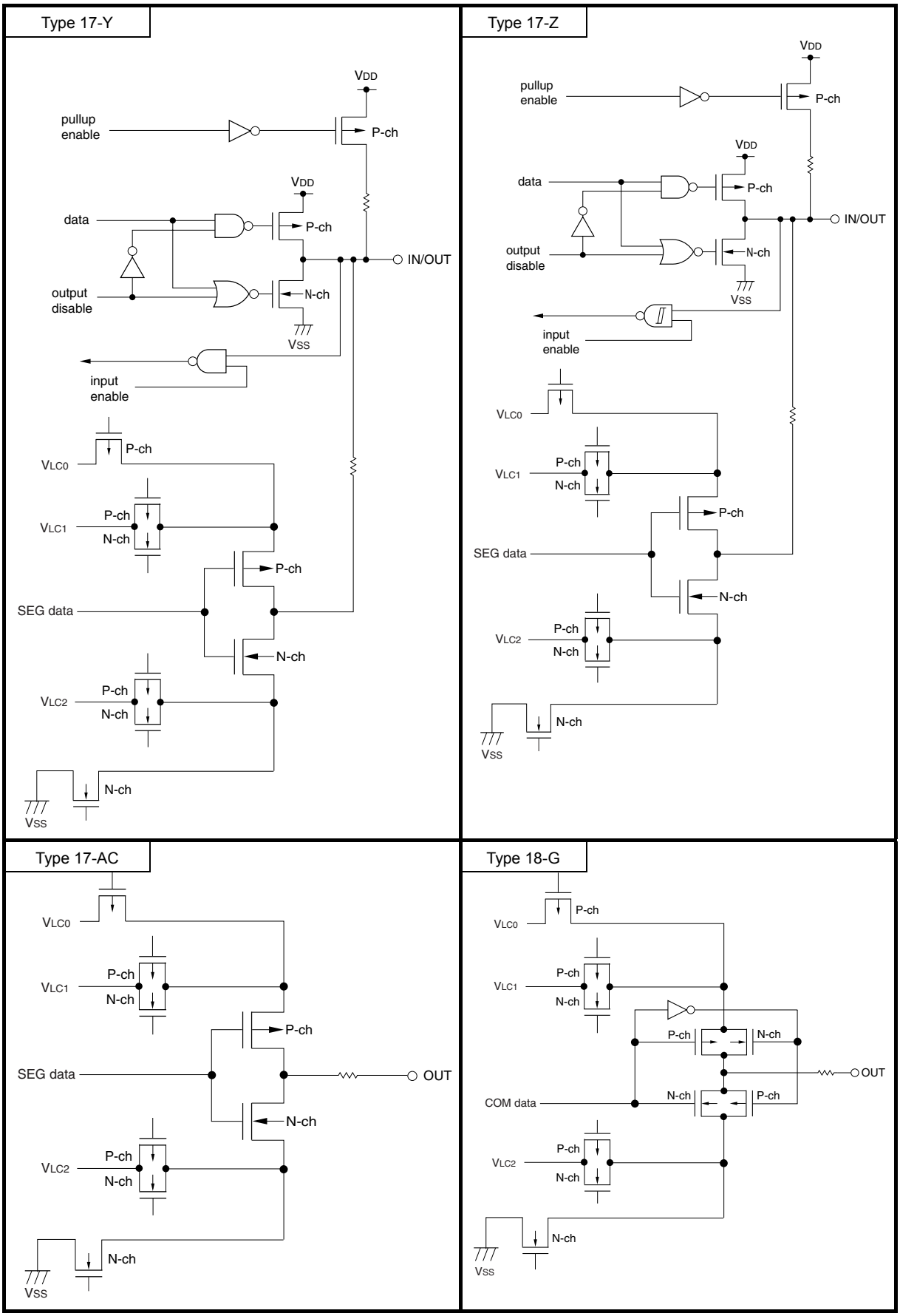
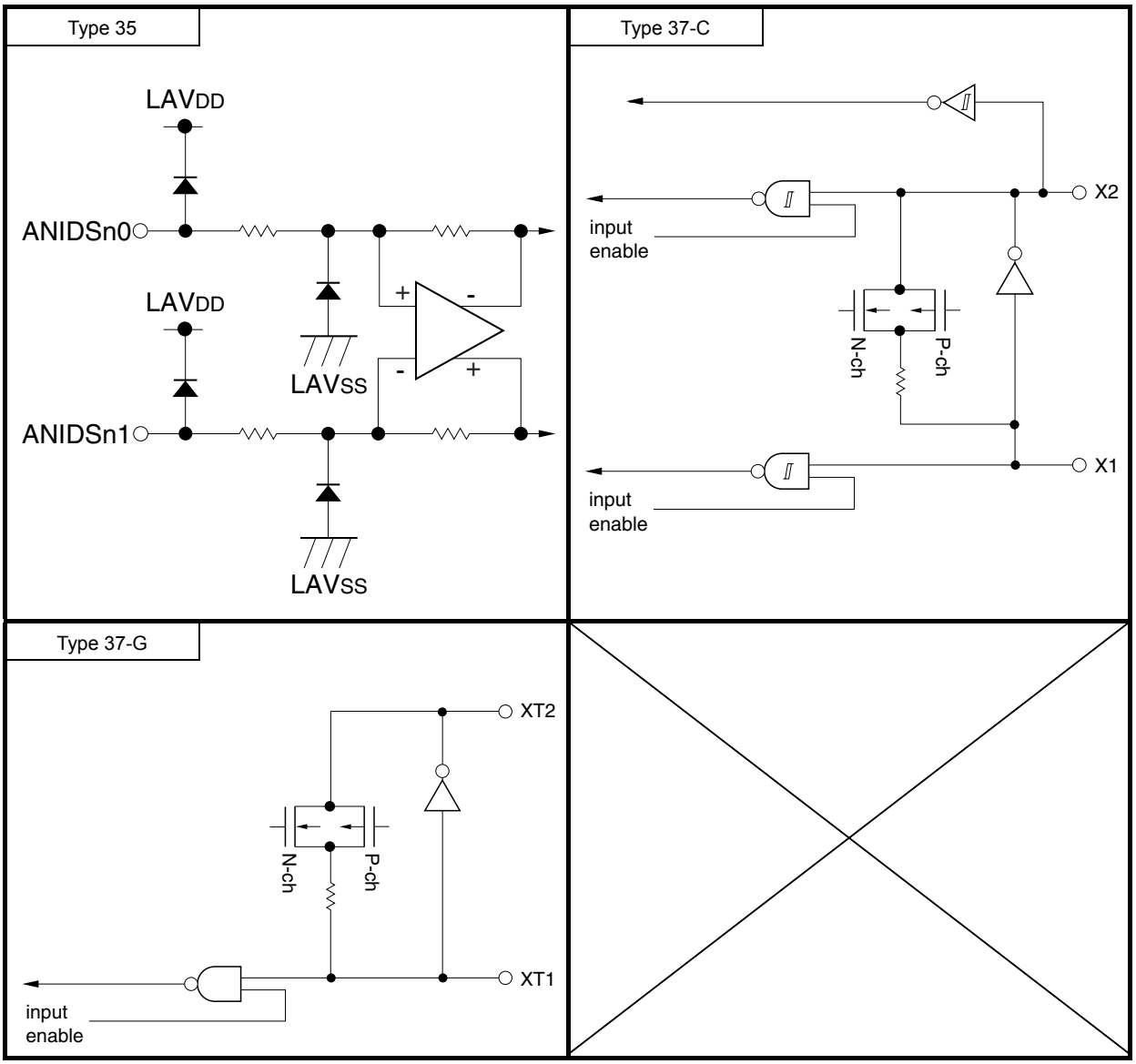


Figure 2-1. Pin I/O Circuit List (4/4)



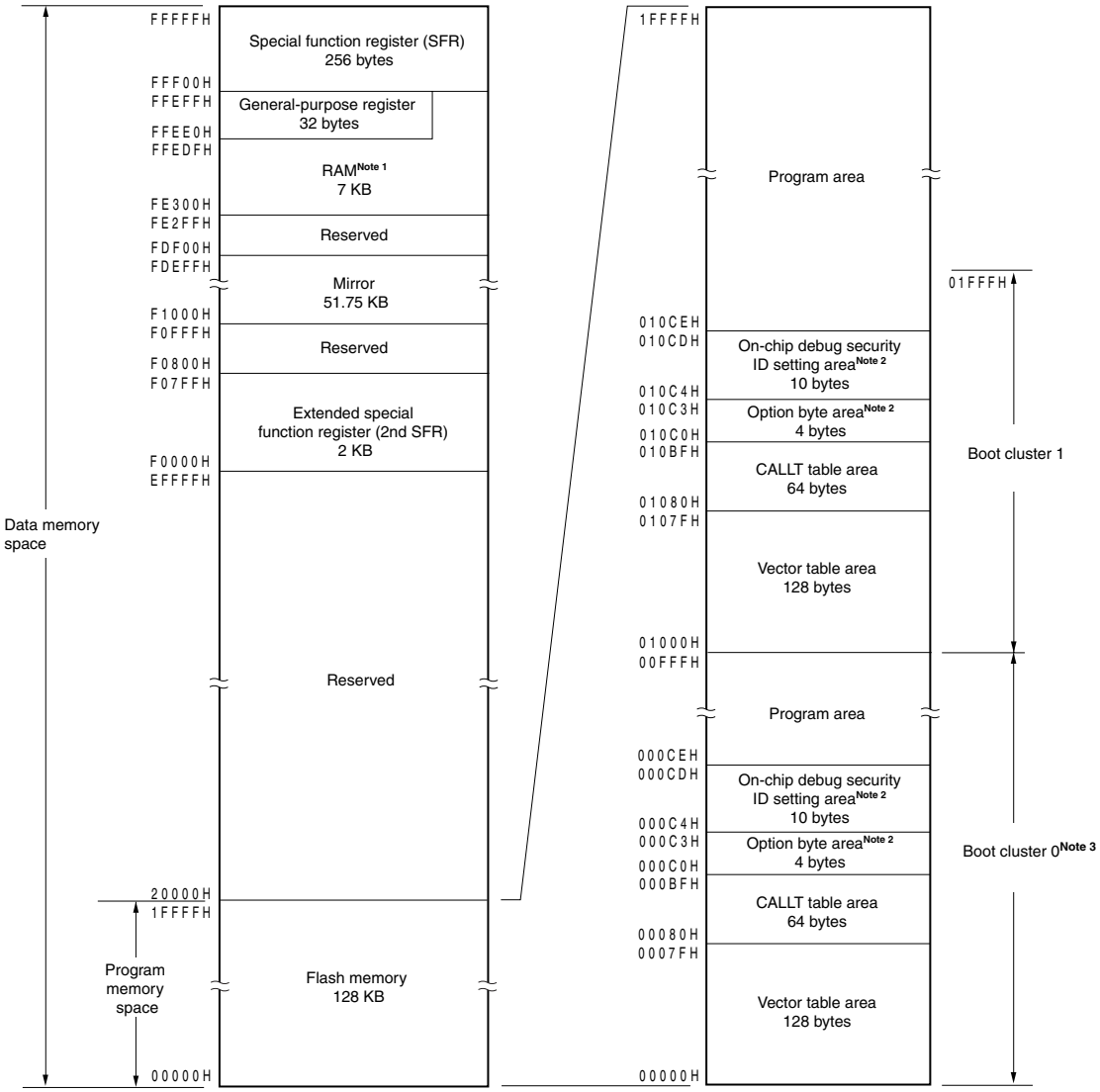
CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

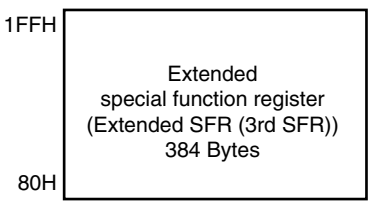
Products in the 78K0R/Lx3-M microcontrollers can access a 1 MB memory space. Figures 3-1 show the memory maps.

Figure 3-1. Memory Map

(a) SFR space



(b) Extended SFR (3rd SFR) space



(**Notes** and **Remark** are listed on the next page.)

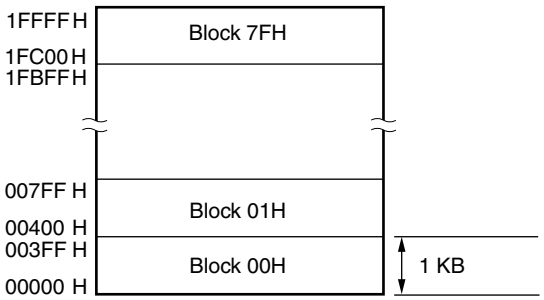
Notes 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

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

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

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.7 Security Setting**).

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

3.1.1 Internal program memory space

The internal program memory space stores the program and table data. Normally, it is addressed with the program counter (PC).

78K0R/Lx3-M microcontrollers products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Internal ROM	
Structure	Capacity
Flash memory	131071 × 8 bits (00000H-1FFFFH)

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 0FFFFH, 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.

Table 3-3. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
00000H	$\overline{\text{RESET}}$ input, POC, LVI, WDT, TRAP	00036H	INTRTC
00004H	INTWDTI	00038H	INTRTCI
00006H	INTLVI	0003CH	INTST2
00008H	INTP0		INTCSI20
0000EH	INTP3		INTIIC20
00014H	INTST3	0003EH	INSR2
00016H	INTSR3	00040H	INTSRE2
00018H	INTSRE3	00042H	INTTM04
0001AH	INTDMA0	00044H	INTTM05
0001CH	INTDMA1	00046H	INTTM06
0001EH	INTST0	00048H	INTTM07
00020H	INTSR0	0004AH	INTP6
00022H	INTSRE0	0004EH	INTP8
00024H	INTCSI10	00050H	INTP9
0002AH	INTIICA	00056H	INTTM10
0002CH	INTTM00	00058H	INTTM11
0002EH	INTTM01	0005AH	INTTM12
00030H	INTTM02	0005CH	INTTM13
00032H	INTTM03	0005EH	INTMD
00034H	INTAD	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

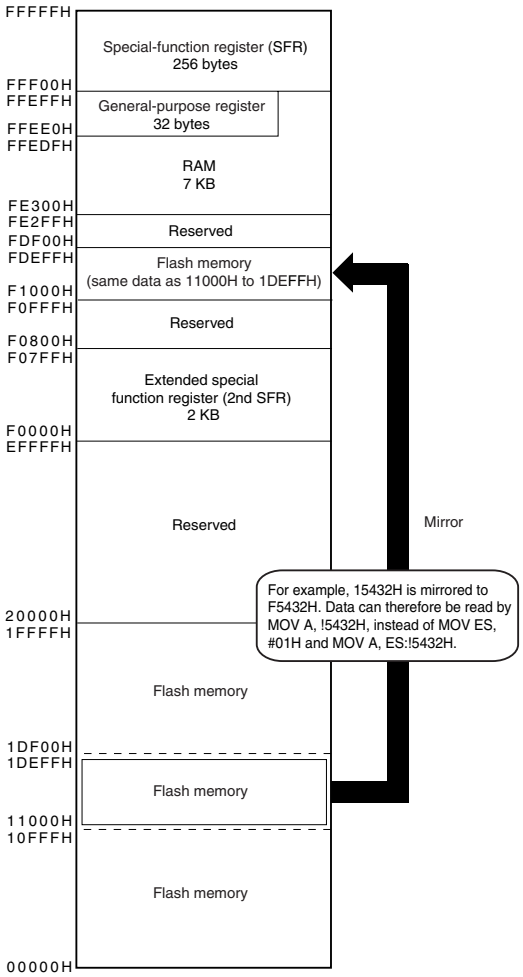
The 78K0R/Lx3-M mirror the data flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, to F0000H to FFFFFH (the data flash area to be mirrored is set by the processor mode control register (PMC)).

By reading data from F0000H to FFFFFH, an instruction that does not have the ES registers as an operand can be used, and thus the contents of the data flash can be read with the shorter code. However, the data flash area is not mirrored to the SFR, extended SFR (2nd SFR), RAM, and use prohibited areas.

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

The following show examples.

Example: Setting MAA = 1



Remark MAA: Bit 0 of the processor mode control register (PMC).

PMC register is described below.

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

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

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

  Reset signal generation sets this register to 00H.

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

Address: FFFFEH After reset: 00H R/W

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

MAA	Selection of flash memory space for mirroring to area from F0000H to FFFFFH
0	00000H to 0FFFFH is mirrored to F0000H to FFFFFH
1	10000H to 1FFFFH is mirrored to F0000H to FFFFFH

Cautions 1. Set PMC only once during the initial settings prior to operating the DMA controller. Rewriting PMC other than during the initial settings is prohibited.

2. After setting PMC, wait for at least one instruction and access the mirror area.

3.1.3 Internal data memory space

78K0R/Lx3-M microcontrollers products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Internal RAM
7168 × 8 bits (FE300H 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 general-purpose registers.

The internal RAM is used as a stack memory.

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

2. While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory.

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FFF00H to FFFFFH.

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.

SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the 2nd SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

Caution Do not access addresses to which 2nd SFRs are not assigned.

3.1.6 Extended special function register (3rd SFR: 3rd Special Function Register) area

On-chip peripheral hardware special function registers (3rd SFRs) are allocated in the area 80H-1D8H (see **Table 3-7** in **3.2.6 Extended Special function registers (3rd SFRs: 3rd Special Function Registers)**).

The extended SFR (3rd SFR) interface is used to access the extended SFR (3rd SFR) space (See **CHAPTER 14 EXTENDED SFR (3rd SFR) INTERFACE**).

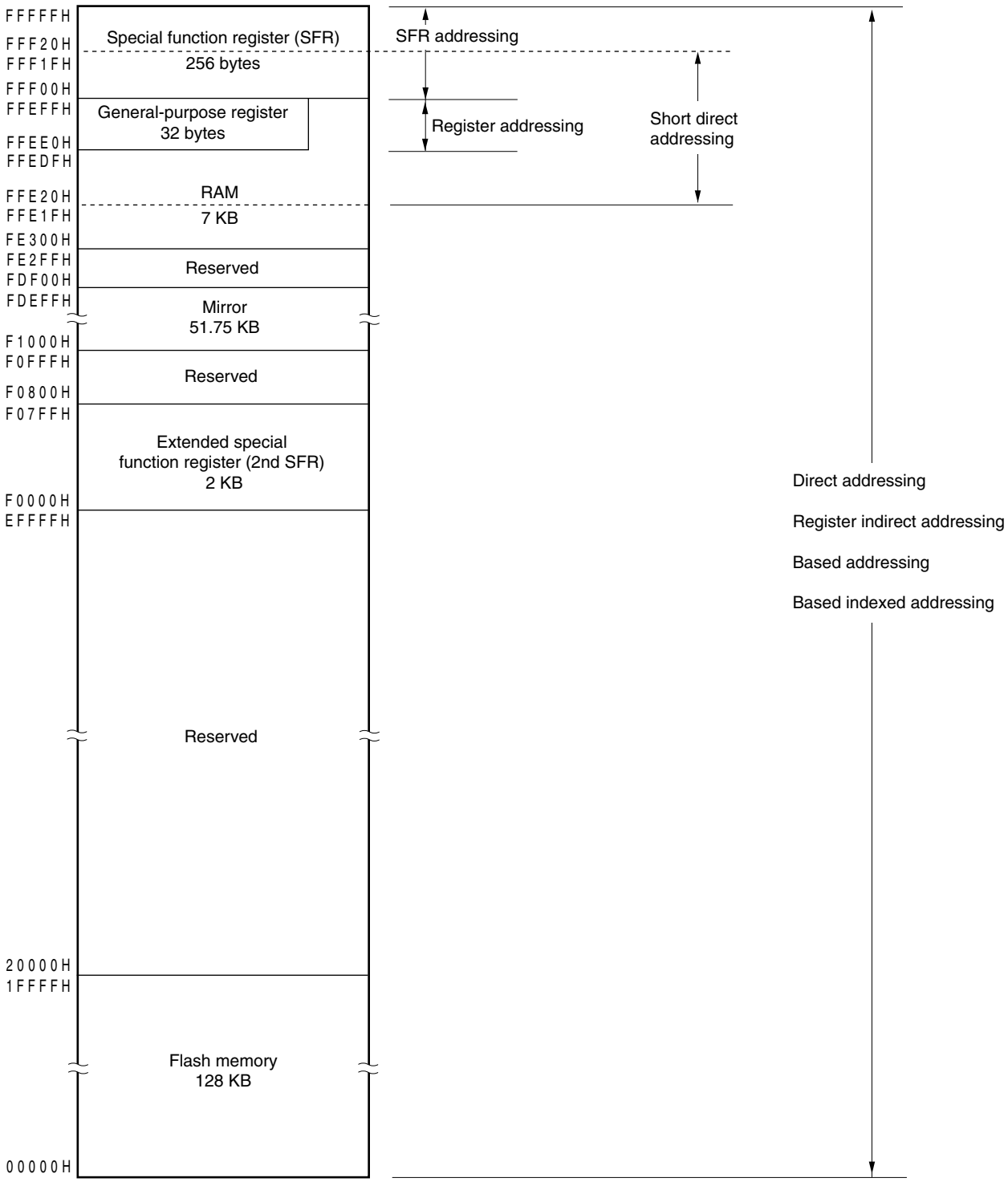
Caution Do not access addresses to which 3rd SFRs are not assigned.

3.1.7 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 78K0R/Lx3-M microcontrollers, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFRs) and general-purpose registers are available for use. Figures 3-3 show correspondence between data memory and addressing.

Figure 3-3. Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The 78K0R/Lx3-M microcontrollers products incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

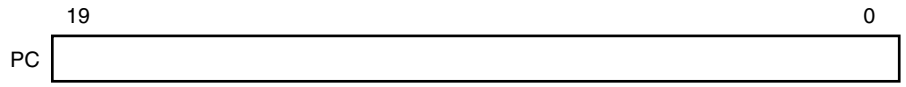
The program counter is a 20-bit register that holds the address information of the next program to be executed.

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

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

Reset signal generation sets the reset vector table values at addresses 0000H and 0001H to the program counter.

Figure 3-4. 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 interrupt request generation or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 3-5. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)

When the operation result is zero, 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 by a priority specification flag register (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see **18.3 (5)**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-6. Format of Stack Pointer

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

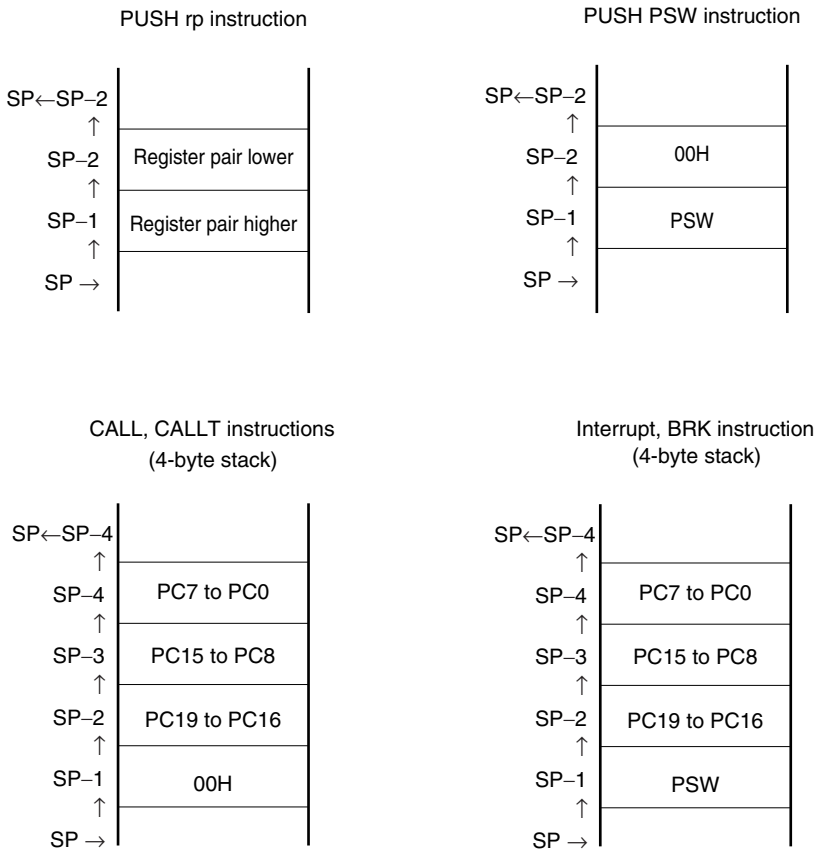
Each stack operation saves data as shown in Figure 3-7.

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 as a stack area.

3. While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory.

Figure 3-7. Data to Be Saved to Stack Memory



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).

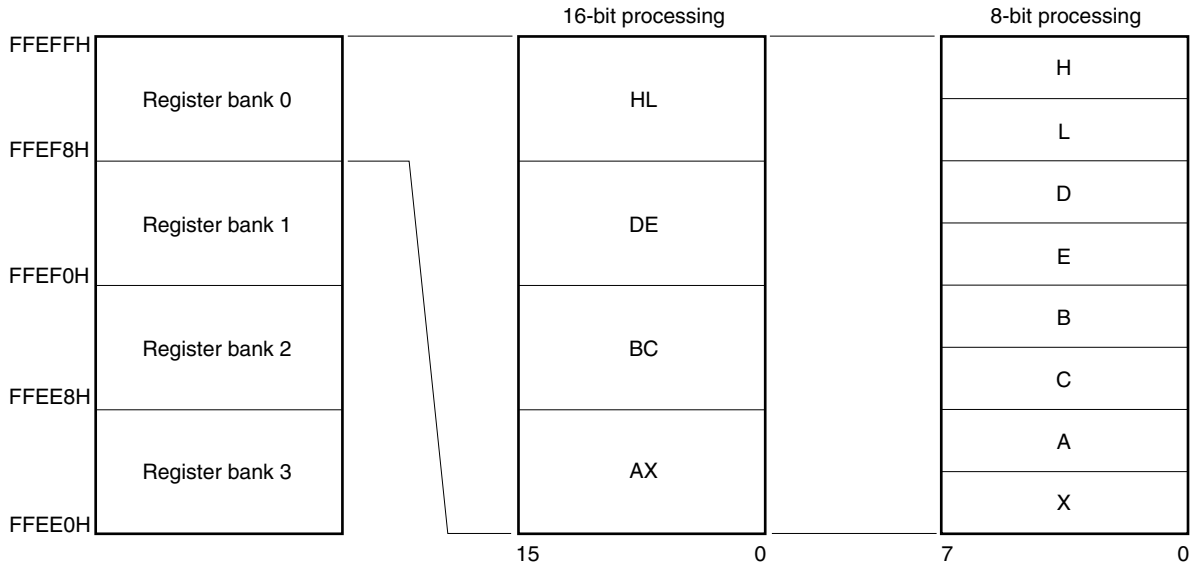
These registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

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 interrupts for each bank.

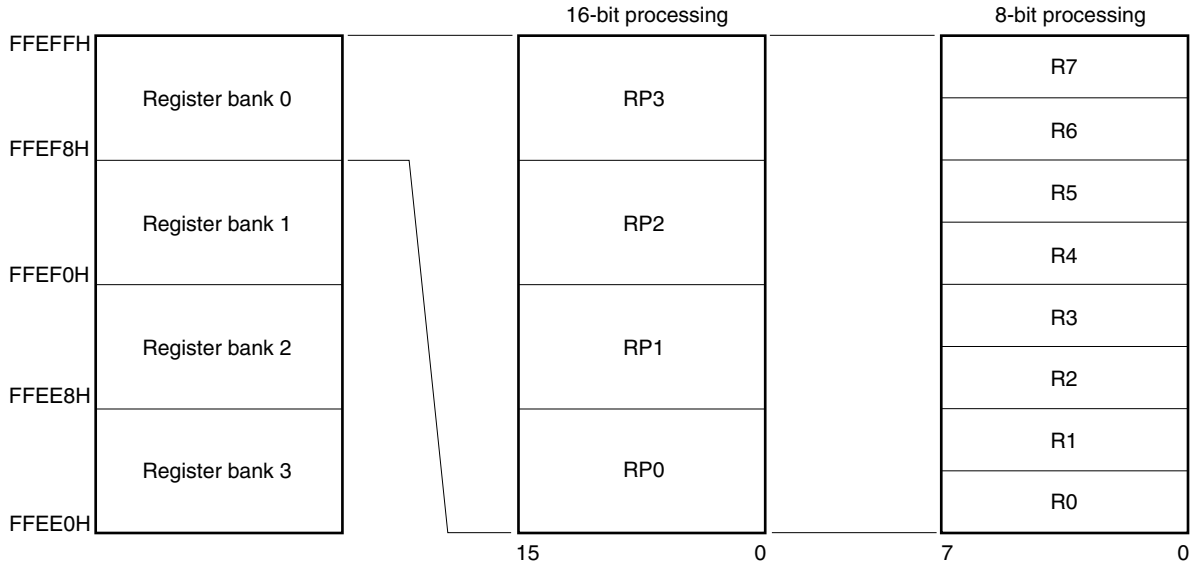
Caution It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

Figure 3-8. Configuration of General-Purpose Registers

(a) Function name



(b) Absolute name



3.2.3 ES and CS registers

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

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

Figure 3-9. Configuration of ES and CS Registers

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

	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CP2	CP1	CP0

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 the symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit). This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (sfr). This manipulation can also be specified with an address.
- 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 RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R. When using the RA78K0R, ID78K0R-QB, and SM+ for 78K0R, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

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

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

For extended SFRs (3rd SFRs), see **3.2.6 Extended special function registers (3rd SFRs: 3rd 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
FFF08H	Port register 8	P8		R/W	√	√	–	00H
FFF09H	Port register 9	P9		R/W	√	√	–	00H
FFF0AH	Port register 10	P10		R/W	√	√	–	00H
FFF0CH	Port register 12	P12		R/W	√	√	–	Undefined
FFF0DH	Port register 13	P13		R/W	√	√	–	00H
FFF0EH	Port register 14	P14		R/W	√	√	–	00H
FFF0FH	Port register 15	P15		R/W	√	√	–	00H
FFF10H	Serial data register 00	TXD0	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	TDR01		R/W	–	–	√	0000H
FFF1BH								
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
FFF28H	Port mode register 8	PM8		R/W	√	√	–	FFH
FFF29H	Port mode register 9	PM9		R/W	√	√	–	FFH

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	
FFF2AH	Port mode register 10	PM10		R/W	√	√	–	FFH
FFF2BH	Port mode register 11	PM11		R/W	√	√	–	FFH
FFF2CH	Port mode register 12	PM12		R/W	√	√	–	FFH
FFF2EH	Port mode register 14	PM14		R/W	√	√	–	FFH
FFF2FH	Port mode register 15	PM15		R/W	√	√	–	FFH
FFF30H	A/D converter mode register	ADM		R/W	√	√	–	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	–	00H
FFF32H	A/D converter mode register 1	ADM1		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
FFF3AH	External interrupt rising edge enable register 1	EGP1		R/W	√	√	–	00H
FFF3BH	External interrupt falling edge enable register 1	EGN1		R/W	√	√	–	00H
FFF3CH	Input switch control register	ISC		R/W	√	√	–	00H
FFF3EH	Timer input select register 0	TIS0		R/W	√	√	–	00H
FFF3FH	Timer input select register 1	TIS1		R/W	√	√	–	00H
FFF40H	LCD mode register	LCDMD		R/W	√	√	–	00H
FFF41H	LCD display mode register	LCDM		R/W	√	√	–	00H
FFF42H	LCD clock control register 0	LCDC0		R/W	√	√	–	00H
FFF43H	LCD boost level control register	VLCD		R/W	√	√	–	0FH
FFF44H	Serial data register 02	SIO10	SDR02	R/W	–	√	√	0000H
FFF45H		–			–	–		
FFF46H	Serial data register 03	RXD1	SDR03	R/W	–	√	√	0000H
FFF47H		–			–	–		
FFF48H	Serial data register 10	TXD2/SIO20	SDR10	R/W	–	√	√	0000H
FFF49H		–			–	–		
FFF4AH	Serial data register 11	RXD2	SDR11	R/W	–	√	√	0000H
FFF4BH		–			–	–		
FFF50H	IICA shift register	IICA		R/W	–	√	–	00H
FFF51H	IICA status register	IICS		R	√	√	–	00H
FFF52H	IICA flag register	IICF		R/W	√	√	–	00H
FFF64H	Timer data register 02	TDR02		R/W	–	–	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03		R/W	–	–	√	0000H
FFF67H								
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								

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	
FFF6EH	Timer data register 07	TDR07	R/W	–	–	√	0000H
FFF6FH							
FFF70H	Timer data register 10	TDR10	R/W	–	–	√	0000H
FFF71H							
FFF72H	Timer data register 11	TDR11	R/W	–	–	√	0000H
FFF73H							
FFF74H	Timer data register 12	TDR12	R/W	–	–	√	0000H
FFF75H							
FFF76H	Timer data register 13	TDR13	R/W	–	–	√	0000H
FFF77H							
FFF90H	Sub-count register	RSUBC	R	–	–	√	0000H
FFF91H							
FFF92H	Second count register	SEC	R/W	–	√	–	00H
FFF93H	Minute count register	MIN	R/W	–	√	–	00H
FFF94H	Hour count register	HOUR	R/W	–	√	–	12H[Note]
FFF95H	Week count register	WEEK	R/W	–	√	–	00H
FFF96H	Day count register	DAY	R/W	–	√	–	01H
FFF97H	Month count register	MONTH	R/W	–	√	–	01H
FFF98H	Year count register	YEAR	R/W	–	√	–	00H
FFF99H	Watch error correction register	SUBCUD	R/W	–	√	–	00H
FFF9AH	Alarm minute register	ALARMWM	R/W	–	√	–	00H
FFF9BH	Alarm hour register	ALARMWH	R/W	–	√	–	12H
FFF9CH	Alarm week register	ALARMWW	R/W	–	√	–	00H
FFF9DH	Real-time counter control register 0	RTCC0	R/W	√	√	–	00H
FFF9EH	Real-time counter control register 1	RTCC1	R/W	√	√	–	00H
FFF9FH	Real-time counter control register 2	RTCC2	R/W	√	√	–	00H
FFFA0H	Clock operation mode control register	CMC	R/W	–	√	–	00H
FFFA1H	Clock operation status control register	CSC	R/W	√	√	–	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC	R	√	√	–	00H
FFFA3H	Oscillation stabilization time select register	OSTS	R/W	–	√	–	07H
FFFA4H	System clock control register	CKC	R/W	√	√	–	09H
FFFA5H	Clock output select register 0	CKS0	R/W	√	√	–	00H

Note The value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) of real-time counter 2 (RTC2) is set to 1 after reset.

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	
FFFA8H	Reset control flag register	RESF		R	–	√	–	Undefined Note 1
FFFA9H	Low-voltage detection register	LVIM		R/W	√	√	–	00H Note 2
FFFAAH	Low-voltage detection level select register	LVIS		R/W	√	√	–	0EH Note 3
FFFABH	Watchdog timer enable register	WDTE		R/W	–	√	–	1A/9A Note 4
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
FFFBAH	DMA mode control register 0	DMC0		R/W	√	√	–	00H
FFFBBH	DMA mode control register 1	DMC1		R/W	√	√	–	00H
FFFBCH	DMA operation control register 0	DRC0		R/W	√	√	–	00H
FFFBDH	DMA operation control register 1	DRC1		R/W	√	√	–	00H
FFFBEH	Back ground event control register	BECTL		R/W	√	√	–	00H
FFFC0H	–	PFCMD Note 5		–	–	–	–	Undefined
FFFC2H	–	PFS Note 5		–	–	–	–	Undefined
FFFC4H	–	FLPMC Note 5		–	–	–	–	Undefined
FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD1H	Interrupt request flag register 2H	IF2H		R/W	√	√		00H
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	Interrupt mask flag register 2H	MK2H		R/W	√	√		FFH
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	Priority specification flag register 02H	PR02H		R/W	√	√		FFH
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH	Priority specification flag register 12H	PR12H		R/W	√	√		FFH
FFFE0H	Interrupt request flag register 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	Interrupt request flag register 0H	IF0H		R/W	√	√		00H
FFFE2H	Interrupt request flag register 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	Interrupt request flag register 1H	IF1H		R/W	√	√		00H

Notes 1. The reset value of RESF varies depending on the reset source.

2. The reset value of LVIM varies depending on the reset source and the setting of the option byte.

3. The reset value of LVIS varies depending on the reset source.

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

5. Do not directly operate this SFR, because it is to be used in the self programming library.

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	
	FFFE4H	Interrupt mask flag register 0L	MK0L	MK0	R/W	√	√	√	FFH
	FFFE5H	Interrupt mask flag register 0H	MK0H		R/W	√	√		FFH
	FFFE6H	Interrupt mask flag register 1L	MK1L	MK1	R/W	√	√	√	FFH
	FFFE7H	Interrupt mask flag register 1H	MK1H		R/W	√	√		FFH
	FFFE8H	Priority specification flag register 00L	PR00L	PR00	R/W	√	√	√	FFH
	FFFE9H	Priority specification flag register 00H	PR00H		R/W	√	√		FFH
	FFFEAH	Priority specification flag register 01L	PR01L	PR01	R/W	√	√	√	FFH
	FFFEBH	Priority specification flag register 01H	PR01H		R/W	√	√		FFH
	FFFECH	Priority specification flag register 10L	PR10L	PR10	R/W	√	√	√	FFH
	FFFEDH	Priority specification flag register 10H	PR10H		R/W	√	√		FFH
	FFFEEH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
	FFFEFH	Priority specification flag register 11H	PR11H		R/W	√	√		FFH
	FFFF0H	Multiplication/division data register A (L)	MDAL/MULA		R/W	–	–	√	0000H
	FFFF1H								
	FFFF2H	Multiplication/division data register A (H)	MDAH/MULB		R/W	–	–	√	0000H
	FFFF3H								
<R>	FFFF4H	Multiplication/division data register B (H)	MDBH/MULOH		R/W	–	–	√	0000H
	FFFF5H								
<R>	FFFF6H	Multiplication/division data register B (L)	MDBL/MULOL		R/W	–	–	√	0000H
	FFFF7H								

Remark For extended SFRs (2nd SFRs), see **Table 3-6 Extended SFR (2nd SFR) List**.
For extended SFRs (3rd SFRs), see **Table 3-7 Extended SFR (3rd SFR) List**.

3.2.5 Extended special function registers (2nd SFRs)

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

Extended SFRs (2nd 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 (2nd SFR) area, however, is 1 byte longer than an instruction that accesses the SFR area.

Extended SFRs (2nd 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 the symbol reserved by the assembler for the 1-bit manipulation instruction operand (!addr16.bit). This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (!addr16). This manipulation can also be specified with an address.
- 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 (2nd SFRs). The meanings of items in the table are as follows.

- Symbol
  Symbol indicating the address of an extended SFR (2nd SFR). It is a reserved word in the RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R. When using the RA78K0R, ID78K0R-QB, and SM+ for 78K0R, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding extended SFR (2nd 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 2nd SFRs are not assigned.

Remark For SFRs in the SFR area, see **3.2.4 Special function registers (SFRs)**.
For extended SFRs (3rd SFRs), see **3.2.6 Extended special function registers (3rd SFRs)**.

Table 3-6. Extended SFR (2nd SFR) List (1/7)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0017H	A/D port configuration register	ADPC		R/W	–	√	–	10H
F0030H	Pull-up resistor option register 0	PU0		R/W	√	√	–	00H
F0031H	Pull-up resistor option register 1	PU1		R/W	√	√	–	00H
F0033H	Pull-up resistor option register 3	PU3		R/W	√	√	–	00H
F0034H	Pull-up resistor option register 4	PU4		R/W	√	√	–	00H
F0035H	Pull-up resistor option register 5	PU5		R/W	√	√	–	00H
F0038H	Pull-up resistor option register 8	PU8		R/W	√	√	–	00H
F0039H	Pull-up resistor option register 9	PU9		R/W	√	√	–	00H
F003AH	Pull-up resistor option register 10	PU10		R/W	√	√	–	00H
F003CH	Pull-up resistor option register 12	PU12		R/W	√	√	–	00H
F003EH	Pull-up resistor option register 14	PU14		R/W	√	√	–	00H
F0041H	Port input mode register 1	PIM1		R/W	√	√	–	00H
F0051H	Port output mode register 1	POM1		R/W	√	√	–	00H
F0058H	Port output mode register 8	POM8		R/W	√	√	–	00H
F0060H	Noise filter enable register 0	NFEN0		R/W	√	√	–	00H
F0061H	Noise filter enable register 1	NFEN1		R/W	√	√	–	00H
F0080H	Port function register	PFALL		R/W	√	√	–	00H
F0081H	Segment enable register	SEGEN		R/W	√	√	–	00H
F00E0H	Multiplication/division data register C (L)	MDCL		R	–	–	√	0000H
F00E1H								
F00E2H	Multiplication/division data register C (H)	MDCH		R	–	–	√	0000H
F00E3H								
F00E8H	Multiplication/division control register	MDUC		R/W	√	√	–	00H
F00F0H	Peripheral enable register 0	PER0		R/W	√	√	–	00H
F00F3H	Operation speed mode control register	OSMC		R/W	–	√	–	00H
F00F4H	Regulator mode control register	RMC		R/W	–	√	–	00H
F00F6H	20 MHz internal high-speed oscillation control register	DSCCTL		R/W	√	√	–	00H
F00FEH	BCD adjust result register	BCDADJ		R	–	√	–	Undefined
F0100H	Serial status register 00	SSR00L	SSR00	R	–	√	√	0000H
F0101H		–			–	–		
F0102H	Serial status register 01	SSR01L	SSR01	R	–	√	√	0000H
F0103H		–			–	–		
F0104H	Serial status register 02	SSR02L	SSR02	R	–	√	√	0000H
F0105H		–			–	–		

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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		–			–	–		
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								
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								
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		–			–	–		
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		–			–	–		

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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		–			–	–		
F014EH	Serial flag clear trigger register 13	SIR13L	SIR13	R/W	–	√	√	0000H
F014FH		–			–	–		
F0150H	Serial mode register 10	SMR10		R/W	–	–	√	0020H
F0151H								
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		–			–	–		
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								

Table 3-6. Extended SFR (2nd SFR) List (4/7)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0186H	Timer counter register 03	TCR03		R	–	–	√	FFFFH
F0187H								
F0188H	Timer counter register 04	TCR04		R	–	–	√	FFFFH
F0189H								
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								
F01A4H	Timer status register 02	TSR02L	TSR02	R	–	√	√	0000H
F01A5H		–			–	–		
F01A8H	Timer status register 04	TSR04L	TSR04	R	–	√	√	0000H
F01A9H		–			–	–		
F01AEH	Timer status register 07	TSR07L	TSR07	R	–	√	√	0000H
F01AFH		–			–	–		

Table 3-6. Extended SFR (2nd SFR) List (5/7)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H
F01B1H		–			–	–		
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	√	√	√	0000H
F01B3H		–			–	–		
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		–			–	–		
F01B6H	Timer clock select register 0	TPS0L	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		–			–	–		
F01C0H	Timer counter register 10	TCR10		R	–	–	√	FFFFH
F01C1H								
F01C2H	Timer counter register 11	TCR11		R	–	–	√	FFFFH
F01C3H								
F01C4H	Timer counter register 12	TCR12		R	–	–	√	FFFFH
F01C5H								
F01C6H	Timer counter register 13	TCR13		R	–	–	√	FFFFH
F01C7H								
F01C8H	Timer mode register 10	TMR10		R/W	–	–	√	0000H
F01C9H								
F01CAH	Timer mode register 11	TMR11		R/W	–	–	√	0000H
F01CBH								
F01CCH	Timer mode register 12	TMR12		R/W	–	–	√	0000H
F01CDH								
F01CEH	Timer mode register 13	TMR13		R/W	–	–	√	0000H
F01CFH								
F01D8H	Timer channel enable status register 1	TE1L	TE1	R	√	√	√	0000H
F01D9H		–			–	–		

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01DAH	Timer channel start register 1	TS1L	TS1	R/W	√	√	√	0000H
F01DBH		–			–	–		
F01DCH	Timer channel stop register 1	TT1L	TT1	R/W	√	√	√	0000H
F01DDH		–			–	–		
F01DEH	Timer clock select register 1	TPS1L	TPS1	R/W	–	√	√	0000H
F01DFH		–			–	–		
F0230H	IICA control register 0	IICCTL0		R/W	√	√	–	00H
F0231H	IICA control register 1	IICCTL1		R/W	√	√	–	00H
F0232H	IICA low-level width setting register	IICWL		R/W	–	√	–	FFH
F0233H	IICA high-level width setting register	IICWH		R/W	–	√	–	FFH
F0234H	Slave address register	SVA		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
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

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0419H	LCD display data memory 25	SEG25	R/W	–	√	–	00H
F041AH	LCD display data memory 26	SEG26	R/W	–	√	–	00H
F041BH	LCD display data memory 27	SEG27	R/W	–	√	–	00H
F041CH	LCD display data memory 28	SEG28	R/W	–	√	–	00H
F041DH	LCD display data memory 29	SEG29	R/W	–	√	–	00H
F041EH	LCD display data memory 30	SEG30	R/W	–	√	–	00H
F041FH	LCD display data memory 31	SEG31	R/W	–	√	–	00H
F0420H	LCD display data memory 32	SEG32	R/W	–	√	–	00H
F0421H	LCD display data memory 33	SEG33	R/W	–	√	–	00H
F0422H	LCD display data memory 34	SEG34	R/W	–	√	–	00H
F0423H	LCD display data memory 35	SEG35	R/W	–	√	–	00H
F0424H	LCD display data memory 36	SEG36	R/W	–	√	–	00H
F0425H	LCD display data memory 37	SEG37	R/W	–	√	–	00H
F0426H	LCD display data memory 38	SEG38	R/W	–	√	–	00H
F0427H	LCD display data memory 39	SEG39	R/W	–	√	–	00H

Remark For SFRs in the SFR area, see **Table 3-5 SFR List**.
For extended SFRs (3rd SFRs), see **Table 3-7 Extended SFR (3rd SFR) List**.

3.2.6 Extended special function registers (3rd SFRs)

Unlike a general-purpose register, each extended SFR (3rd SFR) has a special function.

3rd SFRs are allocated to the 80H to 1D8H areas.

The extended SFR (3rd SFR) interface is used to access the extended SFR (3rd SFR) space (See **CHAPTER 14 EXTENDED SFR (3rd SFR) INTERFACE**).

Table 3-7 gives a list of the extended SFRs (3rd SFRs). The meanings of items in the table are as follows.

- Symbol
  Symbol indicating the address of an extended SFR (3rd SFR).
- R/W
  Indicates whether the corresponding extended special function register can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulatable bit units
  Indicates the manipulatable 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 (3rd SFRs) are not assigned.

Remark For the SFRs in the SFR area, see **3.2.4 Special function registers (SFRs)**.
For extended SFRs (2nd SFRs), see **3.2.5 Extended special function registers (2nd SFRs).**

Table 3-7. Extended SFR (3rd SFR) List (1/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1-Bit	8-Bits	16-Bits	
80H	Extended SFR (3rd SFR) interrupt request flag register 20	IF20		R/W	–	–	–	00H
81H	Extended SFR (3rd SFR) interrupt request flag register 21	IF21		R/W	–	–	–	00H
82H	Extended SFR (3rd SFR) interrupt request flag register 22	IF22		R/W	–	–	–	00H
83H	Extended SFR (3rd SFR) interrupt request flag register 23	IF23		R/W	–	–	–	00H
84H	Extended SFR (3rd SFR) interrupt mask flag register 20	MK20		R/W	–	–	–	FFH
85H	Extended SFR (3rd SFR) interrupt mask flag register 21	MK21		R/W	–	–	–	FFH
86H	Extended SFR (3rd SFR) interrupt mask flag register 22	MK22		R/W	–	–	–	FFH
87H	Extended SFR (3rd SFR) interrupt mask flag register 23	MK23		R/W	–	–	–	FFH
88H	Pull-down status control register	PUTCTL		R/W	–	–	–	00H
8BH	Port register LP0	LP0		R/W	–	–	–	00H
8CH	Port mode register LP0	LPM0		R/W	–	–	–	FFH
8DH	Pull-up resistor option register LP0	LPU0		R/W	–	–	–	00H
91H	Port function control register	PORTCTL		R/W	–	–	–	00H
92H	Real-time counter control register 2	RTCC2		R/W	–	–	–	00H
93H	Alarm minute register	ALARMWM		R/W	–	–	–	00H
94H	Alarm hour register	ALARMWH		R/W	–	–	–	12H
95H	Alarm week register	ALARMWW		R/W	–	–	–	00H
96H	Real-time counter control register 0	RTCC0		R/W	–	–	–	00H
97H	Real-time counter control register 1	RTCC1		R/W	–	–	–	00H
98H	Sub-count register	RSUBC	RSUBCL	R	–	–	–	00H
99H			RSUBCH	R	–	–	–	00H
9AH	Second count register	SEC		R/W	–	–	–	00H
9BH	Minute count register	MIN		R/W	–	–	–	00H
9CH	Hour count register	HOUR		R/W	–	–	–	12H
9DH	Week count register	WEEK		R/W	–	–	–	00H
9EH	Day count register	DAY		R/W	–	–	–	01H
9FH	Month count register	MONTH		R/W	–	–	–	01H
A0H	Year count register	YEAR		R/W	–	–	–	00H
A1H	Watch error correction register	SUBCUD		R/W	–	–	–	00H
A3H	24-bit ΔΣ-type A/D converter mode register	ADM2		R/W	–	–	–	00H
A4H	A/D clock delay setting register	ADLY		R/W	–	–	–	00H
A5H	High pass filter control register 0	HPFC0		R/W	–	–	–	00H
A6H	High pass filter control register 1	HPFC1		R/W	–	–	–	00H
A7H	24-bit ΔΣ-type A/D conversion result register 0	ADCR0	ADCR0L	R	–	–	–	00H
A8H			ADCR0M	R	–	–	–	00H
A9H			ADCR0H	R	–	–	–	00H
AAH	24-bit ΔΣ-type A/D conversion result register 1	ADCR1	ADCR1L	R	–	–	–	00H
ABH			ADCR1M	R	–	–	–	00H
ACH			ADCR1H	R	–	–	–	00H

Remark The 80H to 1D8H registers are accessed by using the extended SFR (3rd SFR) interface.

Table 3-7. Extended SFR (3rd SFR) List (2/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1-Bit	8-Bits	16-Bits	
ADH	24-bit ΔΣ-type A/D conversion result register 2	ADCR2	ADCR2L	R	–	–	–	00H
AEH			ADCR2M	R	–	–	–	00H
AFH			ADCR2H	R	–	–	–	00H
B0H	24-bit ΔΣ-type A/D conversion result register 3	ADCR3	ADCR3L	R	–	–	–	00H
B1H			ADCR3M	R	–	–	–	00H
B2H			ADCR3H	R	–	–	–	00H
B3H	Phase control registers 0	PHC0	PHC0L	R/W	–	–	–	00H
B4H			PHC0H	R/W	–	–	–	00H
B5H	Phase control registers 1	PHC1	PHC1L	R/W	–	–	–	00H
B6H			PHC1H	R/W	–	–	–	00H
CBH	Real-time counter mode register	RTCMD		R/W	–	–	–	–
100H	Period and frequency measurement result register	PFVAL	PFVALL	R	–	–	–	00H
101H			PFVALH	R	–	–	–	00H
102H	Zero-crossing timeout specification register for voltage channel 1	ZXTOUT1	ZXTOUT1L	R/W	–	–	–	FFH
103H			ZXTOUT1H	R/W	–	–	–	03H
104H	Zero-crossing timeout specification register for voltage channel 2	ZXTOUT2	ZXTOUT2L	R/W	–	–	–	FFH
105H			ZXTOUT2H	R/W	–	–	–	03H
106H	SAG line cycle number specification register for voltage channel 1	SAGNUM1		R/W	–	–	–	–
107H	SAG level specification register for voltage channel 1	SAGVAL1	SAGVAL1L	R/W	–	–	–	00H
108H			SAGVAL1M	R/W	–	–	–	00H
109H			SAGVAL1H	R/W	–	–	–	00H
10AH	SAG line cycle number specification register for voltage channel 2	SAGNUM2		R/W	–	–	–	–
10BH	SAG level specification register for voltage channel 2	SAGVAL2	SAGVAL2L	R/W	–	–	–	00H
10CH			SAGVAL2M	R/W	–	–	–	00H
10DH			SAGVAL2H	R/W	–	–	–	00H
10EH	Peak current level specification register	IPKLMT	IPKLMTL	R/W	–	–	–	FFH
<R> 10FH			IPKLMTM	R/W	–	–	–	FFH
110H	Peak voltage level specification register	VPKLMT	VPKLMTL	R/W	–	–	–	FFH
<R> 111H			VPKLMTM	R/W	–	–	–	FFH
112H	Peak current value register	IMAX	IMAXL	R	–	–	–	00H
113H			IMAXM	R	–	–	–	00H
114H			IMAXH	R	–	–	–	00H
116H	Peak current value clearing register	RSTIMAX	RSTIMAXL	R	–	–	–	00H
117H			RSTIMAXM	R	–	–	–	00H
118H			RSTIMAXH	R	–	–	–	00H
119H	Peak voltage value register	VMAX	VMAXL	R	–	–	–	00H
11AH			VMAXM	R	–	–	–	00H
11BH			VMAXH	R	–	–	–	00H
11DH	Peak voltage value clearing register	RSTVMAX	RSTVMAXL	R	–	–	–	00H
11EH			RSTVMAXM	R	–	–	–	00H
11FH			RSTVMAXH	R	–	–	–	00H

Remark The 80H to 1D8H registers are accessed by using the extended SFR (3rd SFR) interface.

Table 3-7. Extended SFR (3rd SFR) List (3/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1-Bit	8-Bits	16-Bits	
120H	Gain specification register	IMATGAIN	IMATGAINL	R/W	–	–	–	00H
121H			IMATGAINH	R/W	–	–	–	00H
122H	Fault detection control register	PQMCTL		R/W	–	–	–	–
123H	Fault detection threshold value specification register	IST		R/W	–	–	–	–
124H	Fault control register	ICHK		R/W	–	–	–	–
150H	Frequency conversion control register	CFCTL		R/W	–	–	–	–
151H	Frequency scaling specification register	CFMUL	CFMULL	R/W	–	–	–	FFH
152H			CFMULH	R/W	–	–	–	FFH
153H	Pulse width specification register	PULCTL		R/W	–	–	–	–
180H	Power calculation mode control register 1	PWCTL1		R/W	–	–	–	–
181H	Power calculation mode control register 2	PWCTL2		R/W	–	–	–	–
182H	No-load level control register	NLCTL		R/W	–	–	–	–
183H	Active power scaling specification register	ACTDIV		R/W	–	–	–	–
184H	Reactive power scaling specification register	READIV		R/W	–	–	–	–
185H	Apparent power scaling specification register	APPDIV		R/W	–	–	–	–
186H	RMS register for voltage channel 1	V1RMS	V1RMSL	R	–	–	–	00H
187H			V1RMSM	R	–	–	–	00H
188H			V1RMSH	R	–	–	–	00H
189H	RMS register for voltage channel 2	V2RMS	V2RMSL	R	–	–	–	00H
18AH			V2RMSM	R	–	–	–	00H
18BH			V2RMSH	R	–	–	–	00H
18CH	RMS register for current channel 1	I1RMS	I1RMSL	R	–	–	–	00H
18DH			I1RMSM	R	–	–	–	00H
18EH			I1RMSH	R	–	–	–	00H
18FH	RMS register for current channel 2	I2RMS	I2RMSL	R	–	–	–	00H
190H			I2RMSM	R	–	–	–	00H
191H			I2RMSH	R	–	–	–	00H
192H	Active power accumulation reading register	ACTHR	ACTHRL	R	–	–	–	00H
193H			ACTHRM	R	–	–	–	00H
194H			ACTHRH	R	–	–	–	00H
196H	Active power accumulation reading and resetting register	RACTHR	RACTHRL	R	–	–	–	00H
197H			RACTHRM	R	–	–	–	00H
198H			RACTHRH	R	–	–	–	00H
199H	Active power accumulation synchronous reading register	LACTHR	LACTHRL	R	–	–	–	00H
19AH			LACTHRM	R	–	–	–	00H
19BH			LACTHRH	R	–	–	–	00H
19CH	Reactive power accumulation reading register	REAHR	REAHRL	R	–	–	–	00H
19DH			REAHRM	R	–	–	–	00H
19EH			REAHRH	R	–	–	–	00H

Remark The 80H to 1D8H registers are accessed by using the extended SFR (3rd SFR) interface.

Table 3-7. Extended SFR (3rd SFR) List (4/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1-Bit	8-Bits	16-Bits	
1A0H	Reactive power accumulation reading and resetting register	RREAHR	RREAHRL	R	–	–	–	00H
1A1H			RREAHRM	R	–	–	–	00H
1A2H			RREAHRH	R	–	–	–	00H
1A3H	Reactive power accumulation synchronous reading register	LREAHR	LREAHRL	R	–	–	–	00H
1A4H			LREAHRM	R	–	–	–	00H
1A5H			LREAHRH	R	–	–	–	00H
1A6H	Apparent power accumulation reading register	APPHR	APPHRL	R	–	–	–	00H
1A7H			APPHRM	R	–	–	–	00H
1A8H			APPHRH	R	–	–	–	00H
1AAH	Apparent power accumulation reading and resetting register	RAPPHR	RAPPHRL	R	–	–	–	00H
1ABH			RAPPHRM	R	–	–	–	00H
1ACH			RAPPHRH	R	–	–	–	00H
1ADH	Apparent power accumulation synchronous reading register	LAPPHR	LAPPHRL	R	–	–	–	00H
1AEH			LAPPHRM	R	–	–	–	00H
1AFH			LAPPHRH	R	–	–	–	00H
1B0H	Line cycle number specification register	LINNUM	LINNUML	R/W	–	–	–	FFH
1B1H			LINNUMH	R/W	–	–	–	FFH
1B2H	Active power gain specification register 1	ACT1GAIN	ACT1GAINL	R/W	–	–	–	00H
1B3H			ACT1GAINH	R/W	–	–	–	00H
1B4H	Active power gain specification register 2	ACT2GAIN	ACT2GAINL	R/W	–	–	–	00H
1B5H			ACT2GAINH	R/W	–	–	–	00H
1B6H	Reactive power gain specification register 1	REA1GAIN	REA1GAINL	R/W	–	–	–	00H
1B7H			REA1GAINH	R/W	–	–	–	00H
1B8H	Reactive power gain specification register 2	REA2GAIN	REA2GAINL	R/W	–	–	–	00H
1B9H			REA2GAINH	R/W	–	–	–	00H
1BAH	Apparent power gain specification register 1	APP1GAIN	APP1GAINL	R/W	–	–	–	00H
1BBH			APP1GAINH	R/W	–	–	–	00H
1BCH	Apparent power gain specification register 2	APP2GAIN	APP2GAINL	R/W	–	–	–	00H
1BDH			APP2GAINH	R/W	–	–	–	00H
1BEH	RMS gain specification register for current channel 1	IRMS1GAIN	IRMS1GAINL	R/W	–	–	–	00H
1BFH			IRMS1GAINH	R/W	–	–	–	00H
1C0H	RMS gain specification register for current channel 2	IRMS2GAIN	IRMS2GAINL	R/W	–	–	–	00H
1C1H			IRMS2GAINH	R/W	–	–	–	00H
1C2H	Active power offset specification register 1	ACT1OS	ACT1OSL	R/W	–	–	–	00H
1C3H			ACT1OSH	R/W	–	–	–	00H
1C4H	Active power offset specification register 2	ACT2OS	ACT2OSL	R/W	–	–	–	00H
1C5H			ACT2OSH	R/W	–	–	–	00H
1C6H	Reactive power offset specification register 1	REA1OS	REA1OSL	R/W	–	–	–	00H
1C7H			REA1OSH	R/W	–	–	–	00H
1C8H	Reactive power offset specification register 2	REA2OS	REA2OSL	R/W	–	–	–	00H
1C9H			REA2OSH	R/W	–	–	–	00H

Remark The 80H to 1D8H registers are accessed by using the extended SFR (3rd SFR) interface.

Table 3-7. Extended SFR (3rd SFR) List (5/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1-Bit	8-Bits	16-Bits	
1CAH	RMS offset specification register for current channel 1	I1RMSOS	I1RMSOSL	R/W	–	–	–	00H
1CBH			I1RMSOSH	R/W	–	–	–	00H
1CCH	RMS offset specification register for voltage channel 1	V1RMSOS	V1RMSOSL	R/W	–	–	–	00H
1CDH			V1RMSOSH	R/W	–	–	–	00H
1CEH	RMS offset specification register for current channel 2	I2RMSOS	I2RMSOSL	R/W	–	–	–	00H
1CFH			I2RMSOSH	R/W	–	–	–	00H
1D0H	RMS offset specification register for voltage channel 2	V2RMSOS	V2RMSOSL	R/W	–	–	–	00H
1D1H			V2RMSOSH	R/W	–	–	–	00H
1D2H	Sampling mode selection register	SAMPMODE		R/W	–	–	–	–
1D3H	Sampling result register 1	SAMP1	SAMP1L	R	–	–	–	00H
1D4H			SAMP1M	R	–	–	–	00H
1D5H			SAMP1H	R	–	–	–	00H
1D6H	Sampling result register 2	SAMP2	SAMP2L	R	–	–	–	00H
1D7H			SAMP2M	R	–	–	–	00H
1D8H			SAMP2H	R	–	–	–	00H

Remarks 1. The 80H to 1D8H registers are accessed by using the extended SFR (3rd SFR) interface.

2. For SFRs in the SFR area, see **Table 3-5 SFR List**.
For extended SFRs (2nd SFRs), see **Table 3-6 Extended SFR (2nd SFR) List**.

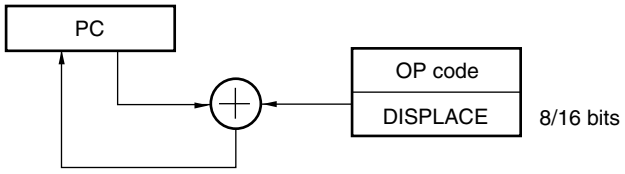
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

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

Figure 3-10. Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

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

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

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

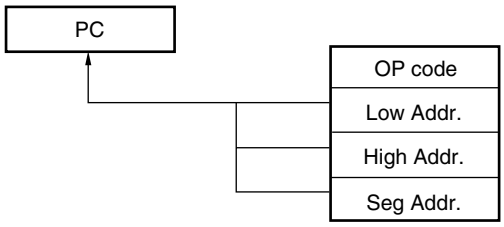
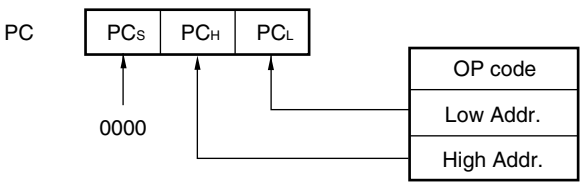


Figure 3-12. Example of CALL !addr16/BR !addr16



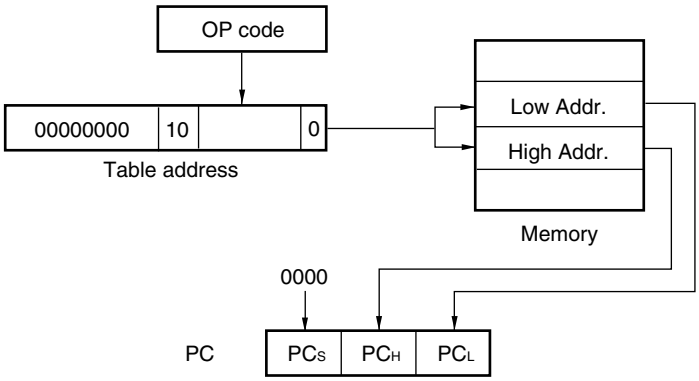
3.3.3 Table indirect addressing

[Function]

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

In the 78K0R microcontrollers, branching is enabled only to the 64 KB space from 00000H to 0FFFFH.

Figure 3-13. Outline of Table Indirect Addressing

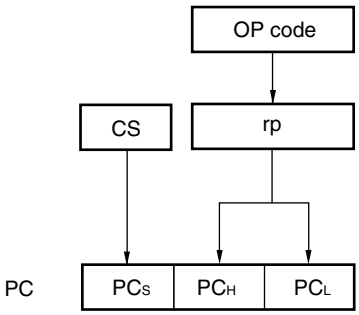


3.3.4 Register direct addressing

[Function]

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

Figure 3-14. Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

[Function]

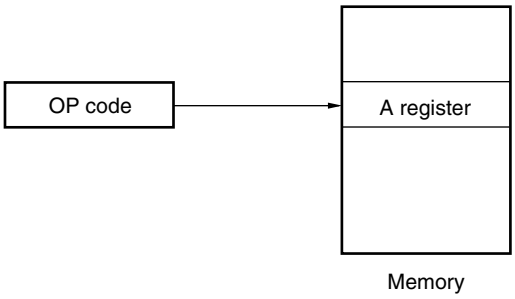
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[Operand format]

Because implied addressing can be automatically employed with an instruction, no particular operand format is necessary.

Implied addressing can be applied only to MULU X.

Figure 3-15. Outline of Implied Addressing



3.4.2 Register addressing

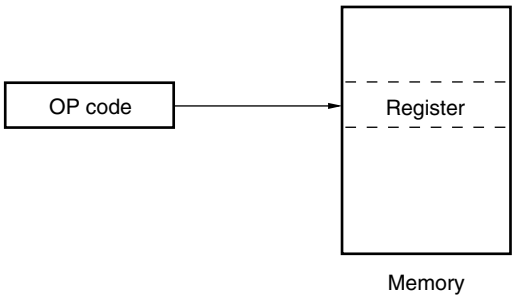
[Function]

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

[Operand format]

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

Figure 3-16. Outline of Register Addressing



3.4.3 Direct addressing

[Function]

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

[Operand format]

Identifier	Description
ADDR16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES: ADDR16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3-17. Example of ADDR16

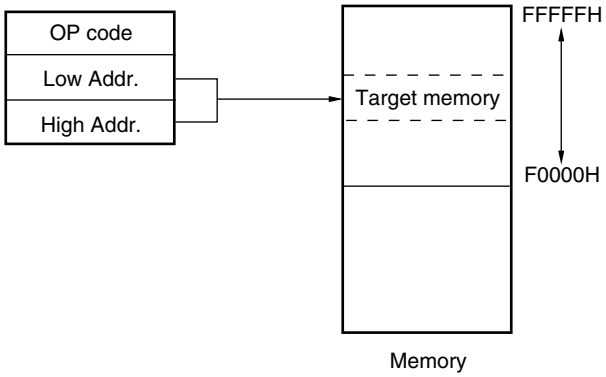
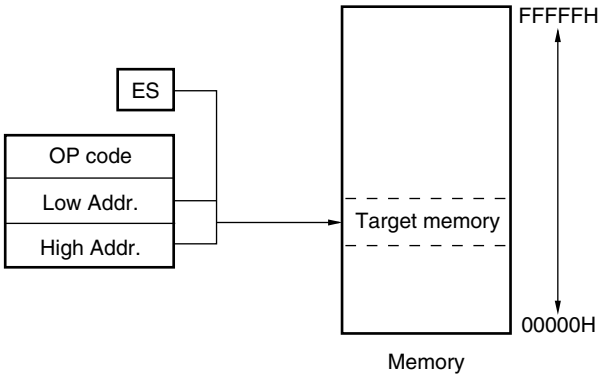


Figure 3-18. Example of ES:ADDR16



3.4.4 Short direct addressing

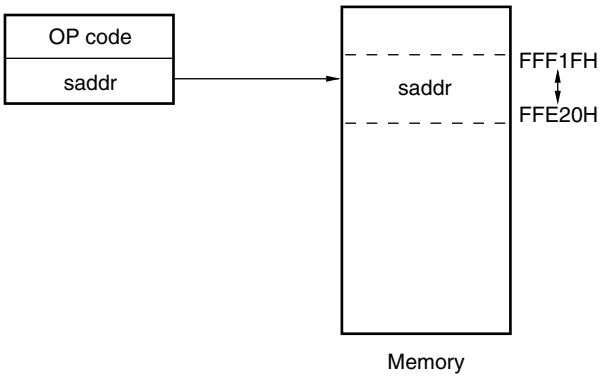
[Function]

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

Figure 3-19. Outline of Short Direct Addressing



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

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

3.4.5 SFR addressing

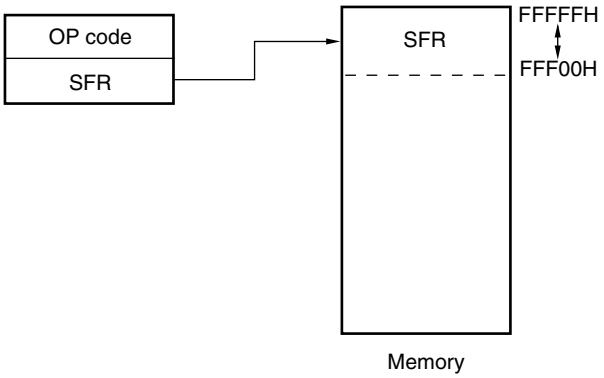
[Function]

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

[Operand format]

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

Figure 3-20. Outline of SFR Addressing



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

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

Figure 3-21. Example of [DE], [HL]

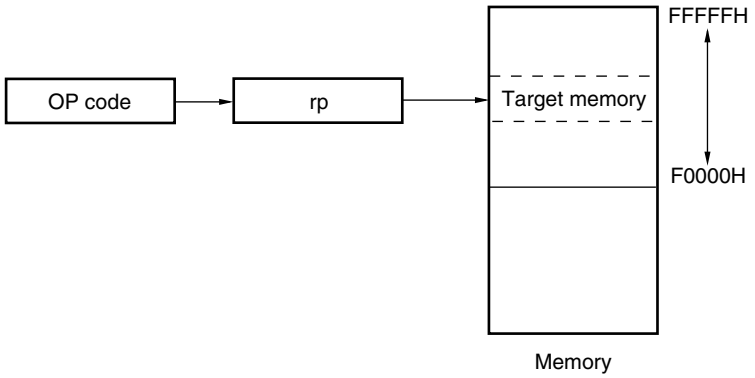
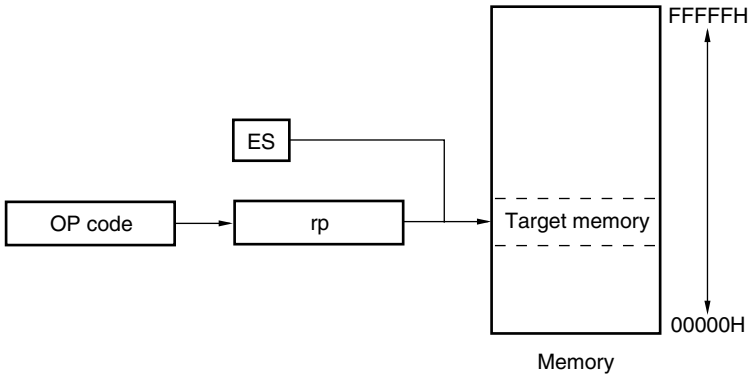


Figure 3-22. Example of ES:[DE], ES:[HL]



3.4.7 Based addressing

[Function]

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

[Operand format]

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

Figure 3-23. Example of [SP+byte]

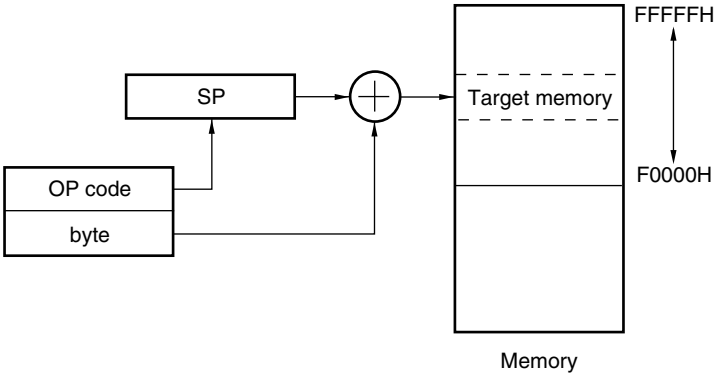


Figure 3-24. Example of [HL + byte], [DE + byte]

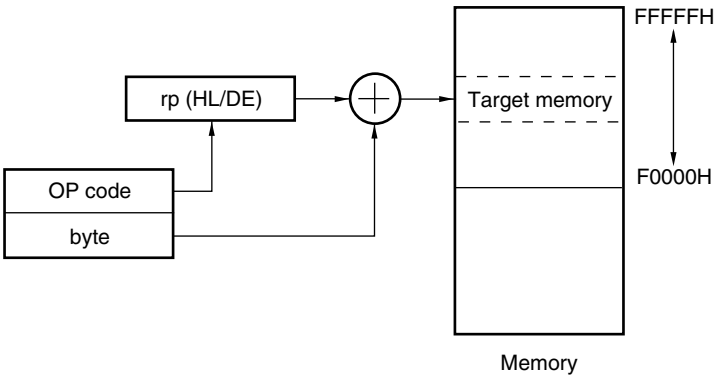


Figure 3-25. Example of word[B], word[C]

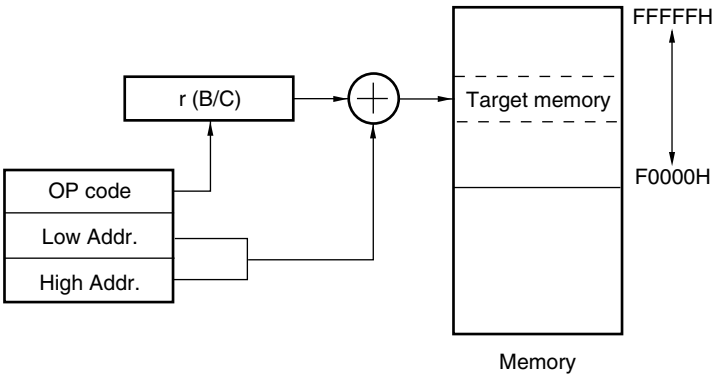


Figure 3-26. Example of word[BC]

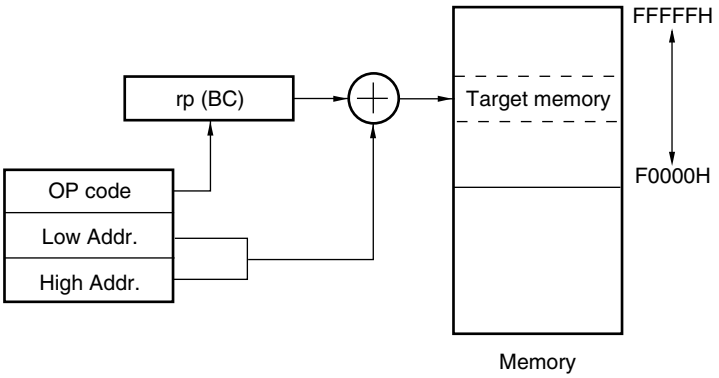


Figure 3-27. Example of ES:[HL + byte], ES:[DE + byte]

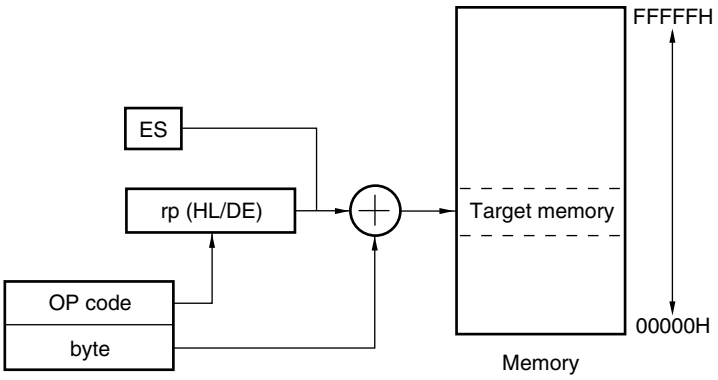


Figure 3-28. Example of ES:word[B], ES:word[C]

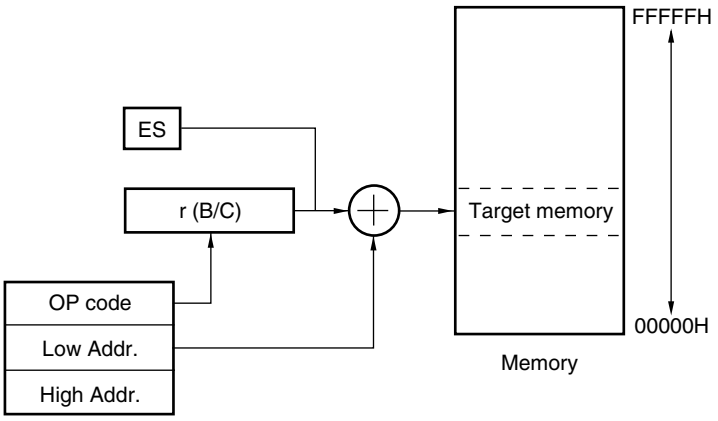
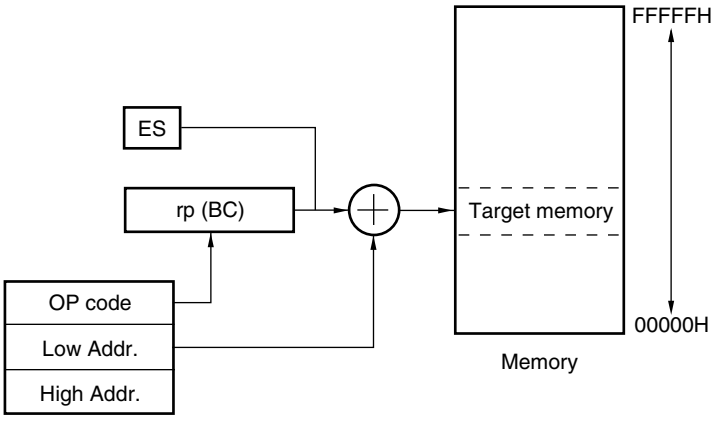


Figure 3-29. Example of ES:word[BC]



3.4.8 Based indexed addressing

[Function]

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

[Operand format]

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

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

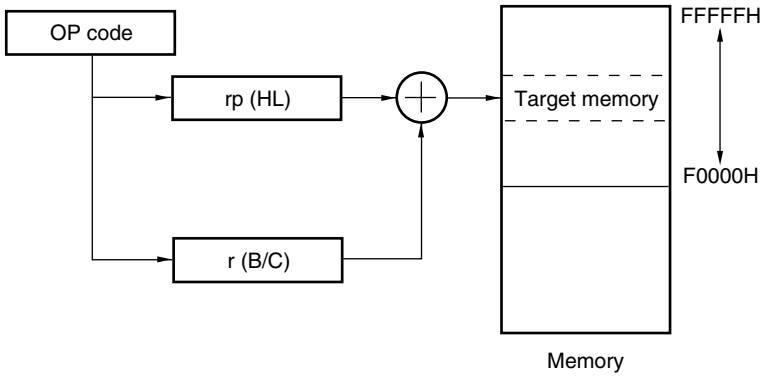
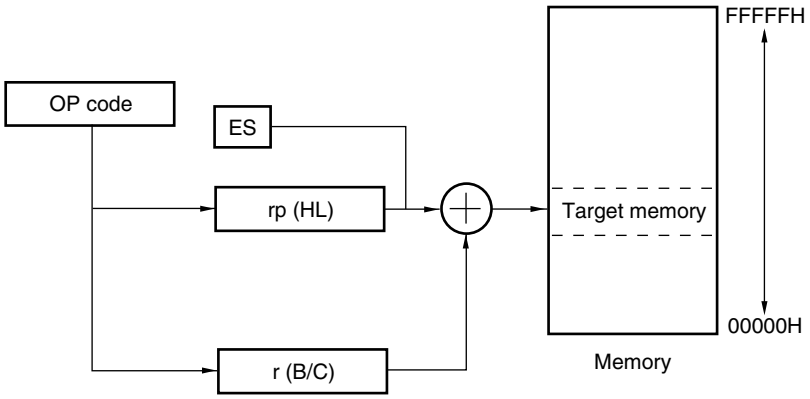


Figure 3-31. Example of ES:[HL+B], ES:[HL+C]



3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) contents. 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.

Stack addressing is applied only to the internal RAM area.

[Operand format]

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

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

There are four types of pin I/O buffer power supplies: AV_{DD}, LAV_{DD}, LV_{DD}, and V_{DD}. The relationship between these power supplies and the pins is shown below.

Table 4-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
AV_{DD}	P152, P157
LAV_{DD}	ANIDS00, ANIDS01, ANIDS10, ANIDS11, ANIDS20, ANIDS21, ANIDS30, ANIDS31 pins
LV_{DD}	CF, EXCLKS1, LP01, XT1, XT2, ZX1, LRESET, FXTOUT, TO05, P16 pins
V_{DD}	Pins other than port

78K0R/Lx3-M products are provided with digital I/O ports, which enable variety of control operations. The functions of each port are shown in Tables 4-2.

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**.

Table 4-2. Port functions (1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	CAPH
P01				CAPL
P10	I/O	Port 1. 3-bit I/O port and 1-bit output port Input/output can be specified in 1-bit units. Input of P10 and P11 can be set to TTL buffer. Output of P10 to P12 can be set to N-ch open-drain output (V_{DD} tolerance). For input of P10 to P12, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{SCK20}$/SCL20
P11				SI20/RxD2/SDA20/ INTP6
P12				SO20/TxD2/TO02
<R> P16	Output		Output port Note 1	TO05
P33	I/O	Port 3. 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI07/TO07/INTP3
P40 Note 2	I/O	Port 4. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOOL0
P41				TOOL1
P50	I/O	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.	Input port	SEG39/RxD3
P51				SEG38/TxD3
P52				SEG37/TI02
P53				SEG36/TI04
P54 to P57				SEG35 to SEG32
P60	I/O	Port 6. 2-bit I/O port. Output is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCL0
P61				SDA0
P81	I/O	Port 8. 2-bit I/O port. Inputs/output can be specified in 1-bit units. Output of P82 can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	RxD0/INTP9
P82				TxD0
P90 to P97	I/O	Port 9. 8-bit I/O port. Inputs/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG31 to SEG24
P100	I/O	Port 10. 1-bit I/O port. Inputs/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG15

<R> **Notes 1.** When using P16/TO05, after reset release, be sure to clear PM16 bit of PM1 and PM20 bit of PM2 to 0, and to set P20 bit of P2 to 1.

2. If on-chip debugging is enabled by using an option byte, be sure to pull up the P40/TOOL0 pin externally.

Table 4-2. Port functions (2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P120	I/O	Port 12. 1-bit I/O port and 3-bit input port. For only P120, input/output can be specified in 1-bit units. For only P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121	Input			X1
P122				X2/EXCLK
P124				EXCLKS0
P140 to P147	I/O	Port 14. 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.	Input port	SEG23 to SEG16
P152	I/O	Port 15. 2-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI10
P157				ANI15
LP01	I/O	Port LP0. 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	ZX1

4.2 Port Configuration

Ports include the following hardware.

Table 4-3. Port Configuration

Item	Configuration	
Control registers	Port mode registers (PMxx, LPM0):	PM0, PM1, PM3 to PM6, PM8 to PM10, PM12, PM14, PM15, LPM0
	Port registers (Pxx, LP0):	P0, P1, P3 to P6, P8 to P10, P12, P14, P15, LP0
	Pull-up resistor option registers (PUxx, LPU0):	PU0, PU1, PU3 to PU5, PU8 to PU10, PU12, PU14, LPU0
	Port input mode registers (PIM1)	
	Port output mode registers (POM1, POM8)	
	A/D port configuration register (ADPC)	
	Port function register (PFALL)	
	Input switch control register (ISC)	
Port	Total: 45 (CMOS I/O: 39, CMOS output: 1, CMOS input: 3, N-ch open drain I/O: 2)	
Pull-up resistor	Total: 37	

4.2.1 Port 0

78K0R/LG3-M (100 pin: μPD78F8070)
P00/CAPH
P01/CAPL

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 and P01 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

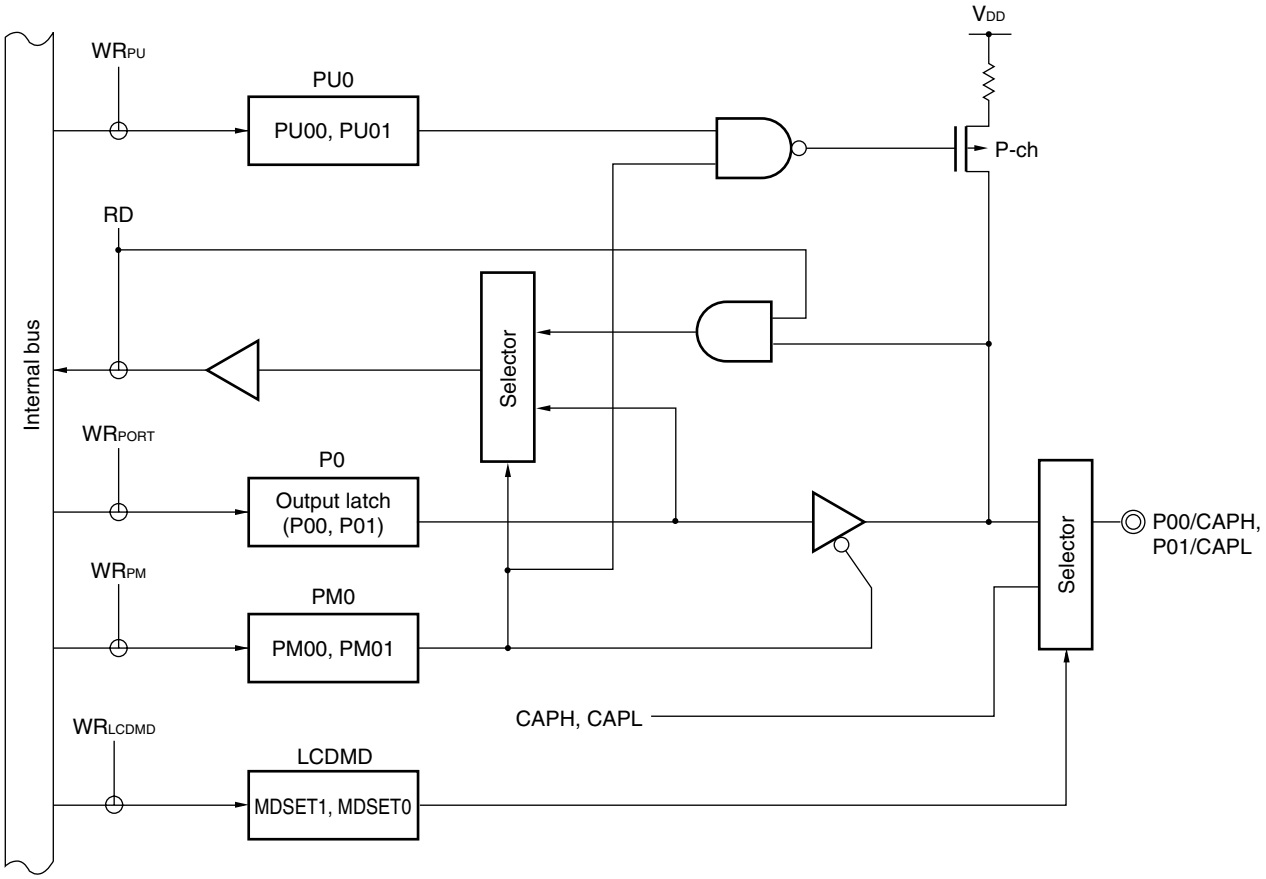
This port can also be used for connecting a capacitor for LCD controller/driver.

Reset signal generation sets port 0 to input mode.

Figures 4-1 show block diagrams of port 0.

Caution To use P00/CAPH and P01/CAPL as a general-purpose port, set bit 5 (MDSET1) and bit 4 (MDSET0) of LCD mode register (LCDMD) to "0", which is the same as their default status setting.

Figure 4-1. Block Diagram of P00 and P01



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
LCDMD: LCD mode register
RD: Read signal
WR××: Write signal

4.2.2 Port 1

78K0R/LG3-M (100pin: μPD78F8070)
P10/$\overline{\text{SCK20}}$/SCL20
P11/SI20/RxD2/SDA20/INTP6
P12/SO20/TxD2/TO02
P16/TO05

P10 to P12 are an 3-bit 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 they 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 P10 and P11 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 P10 to P12 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 1 (POM1).

P16 is only for 1-bit output port.

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

Reset signal generation sets port 1 to input mode.

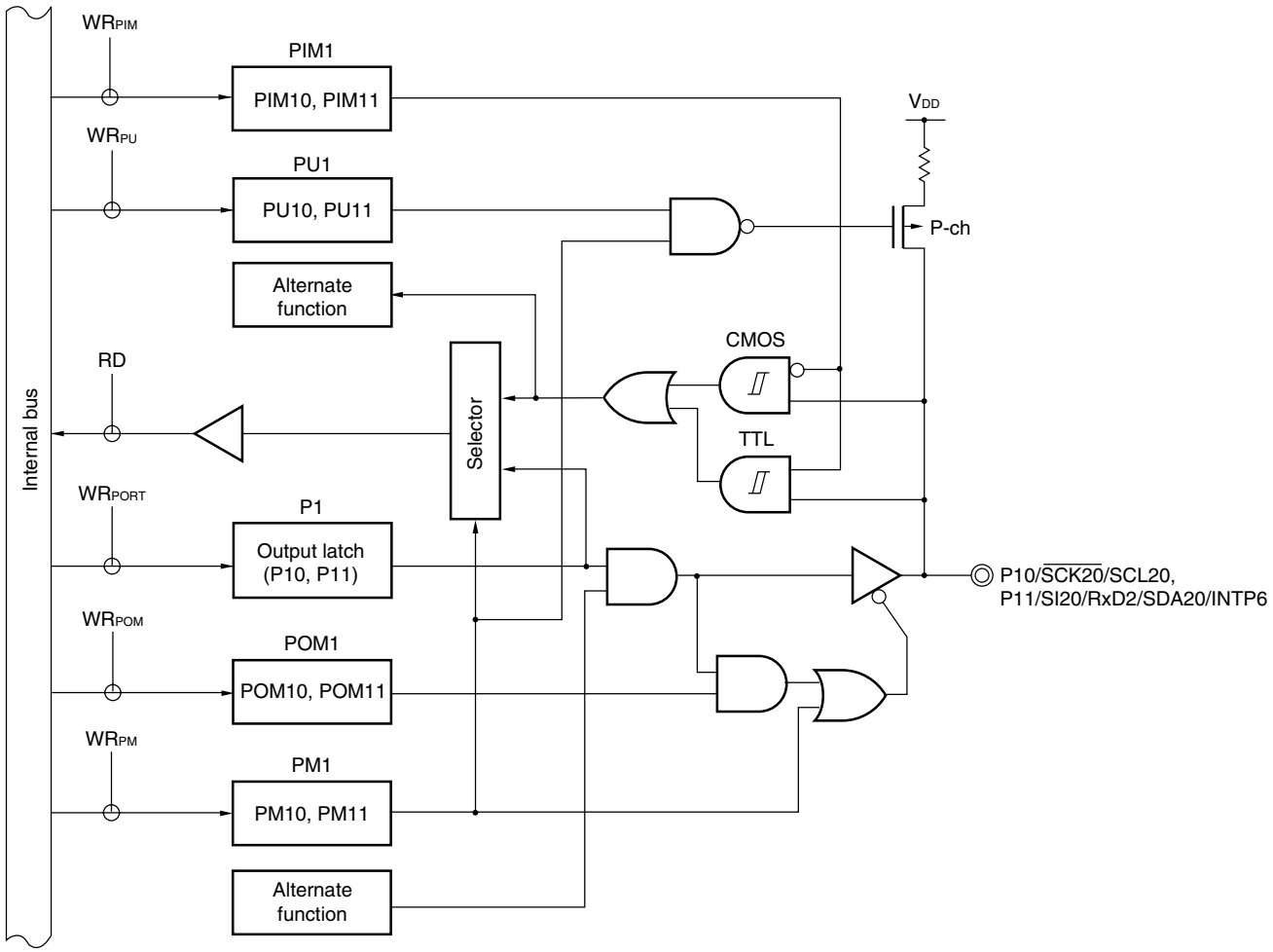
Figures 4-2 to 4-4 show block diagrams of port 1.

Cautions 1. To use P10/$\overline{\text{SCK20}}$/SCL20 and P11/SI20/RxD2/SDA20/INTP6 as a general-purpose port, note the serial array unit 1 setting. For details, refer to Table 12-7 Relationship Between Register Settings and Pins (Channel 0 of unit 1: CSI20, UART2 Reception, IIC20).

2. To use P12/TO02/SO20/TxD2 as a general-purpose port, set bit 2 (TO02) of timer output register 0 (TO0) and bit 2 (TOE02) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting. And as a general-purpose port, note the serial array unit 1 setting. For details of serial array unit 1 setting, refer to Table 12-7 Relationship Between Register Settings and Pins (Channel 0 of unit 1: CSI20, UART2 Reception, IIC20).

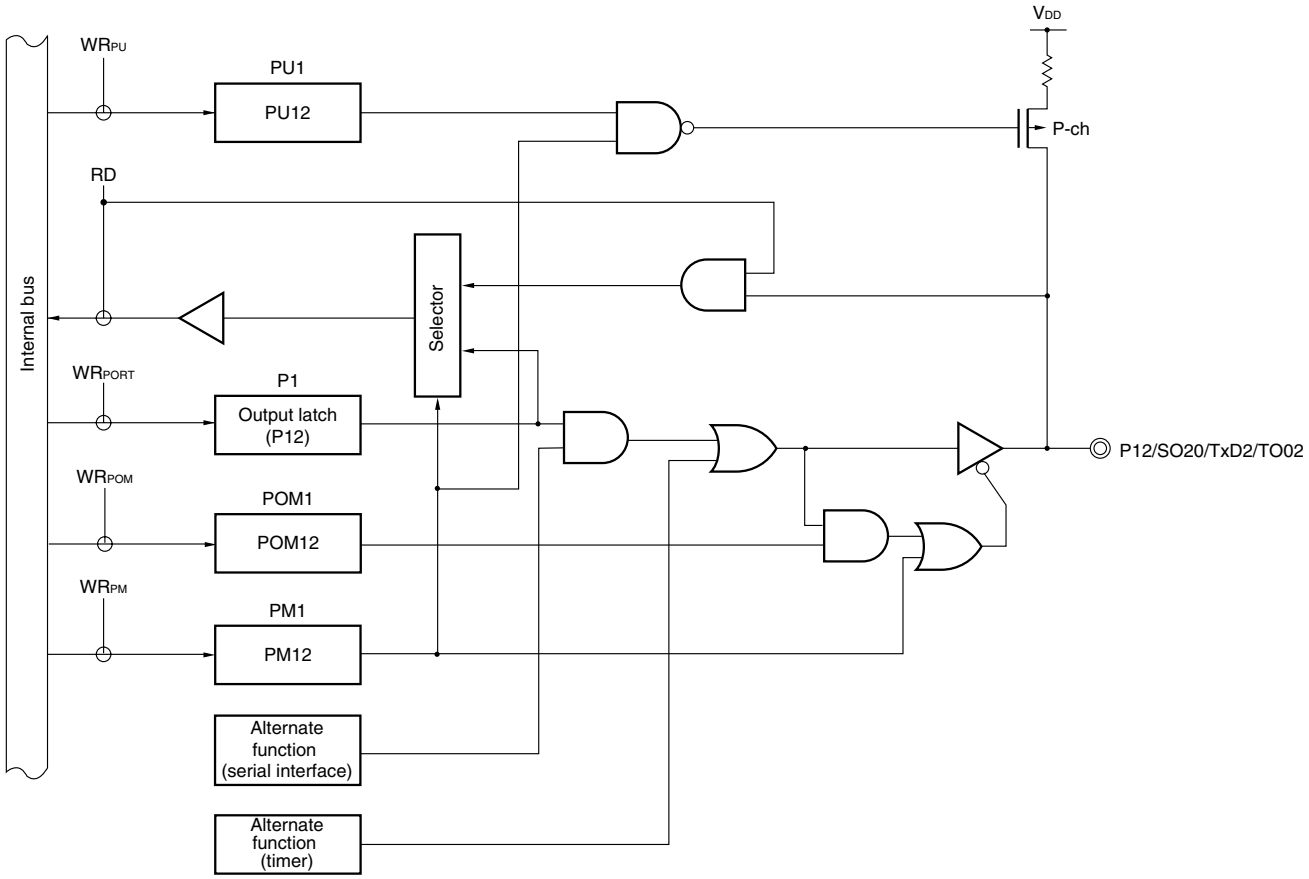
3. To use P16/TO05 as a general-purpose port, set bit 5 (TO05) of timer output register 0 (TO0) and bit 5 (TOE05) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting. Also, be sure to set 0 to the PM20 bit of PM2 after reset is released, and set 1 to the P20 bit of P2 after reset is released.

Figure 4-2. Block Diagram of P10 and P11



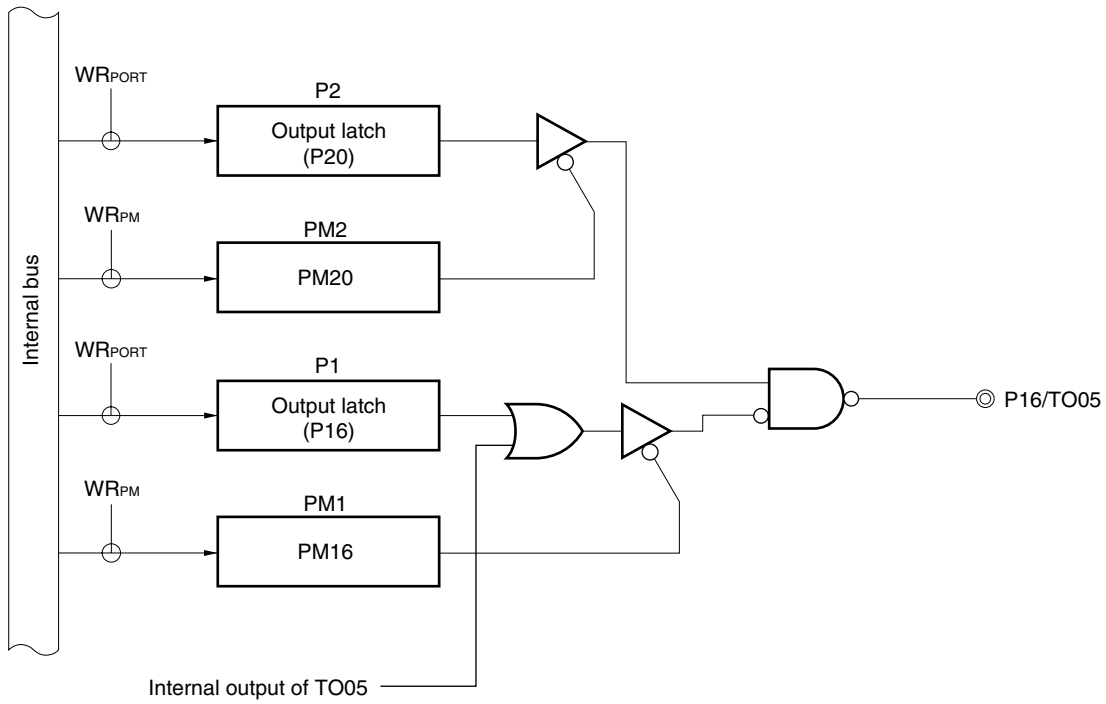
P1: Port register 1
PU1: Pull-up resistor option register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-3. Block Diagram of P12



P1: Port register 1
PU1: Pull-up resistor option register 1
POM1: Port output mode register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-4. Block Diagram of P16



Caution When P16/TO05 is used, be sure to set 0 to the PM20 bit of PM2 after reset is released, and set 1 to the P20 bit of P2 after reset is released.

P1: Port register 1
P2: Port register 2
PM1: Port mode register 1
PM2: Port mode register 2
RD: Read signal
WR××: Write signal

4.2.3 Port 3

78K0R/LG3-M (100 pin: μPD78F8070)
P33/TI07/TO07/INTP3

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 P33 pin is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

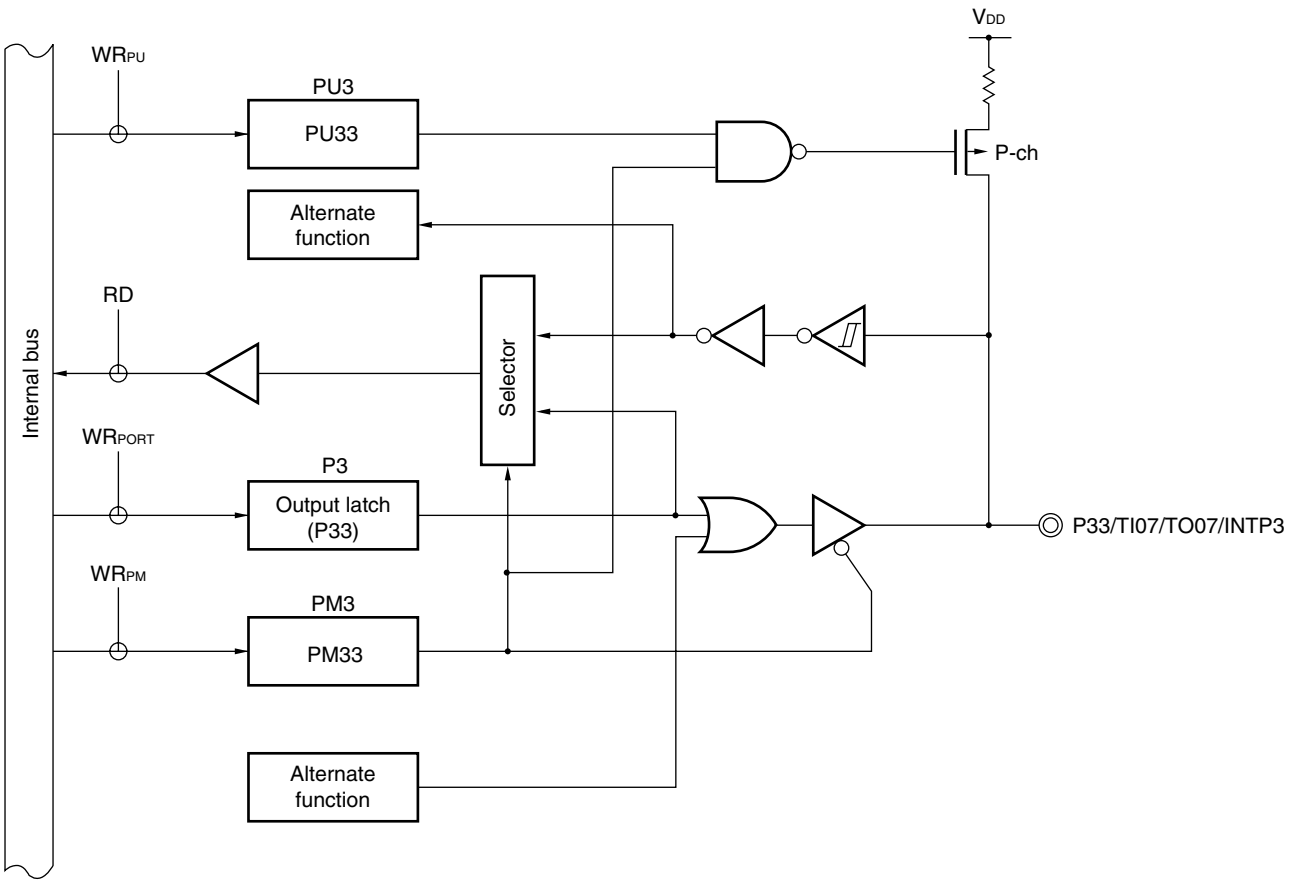
This port can also be used for timer I/O and external interrupt request input.

Reset signal generation sets port 3 to input mode.

Figure 4-5 shows a block diagram of port 3.

Caution To use P33/TO07/TI07/INTP3 as a general-purpose port, set bit 7 (TO07) of timer output register 0 (TO0), and bit 7 (TOE07) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

Figure 4-5. Block Diagram of P33



P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

4.2.4 Port 4

78K0R/LG3-M (100 pin: μPD78F8070)
P40/TOOL0
P41/TOOL1

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 and P41 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 2 (PU2)[Note].

This port can also be used for flash memory programmer/debugger data I/O and debugger clock output.

Reset signal generation sets port 4 to input mode.

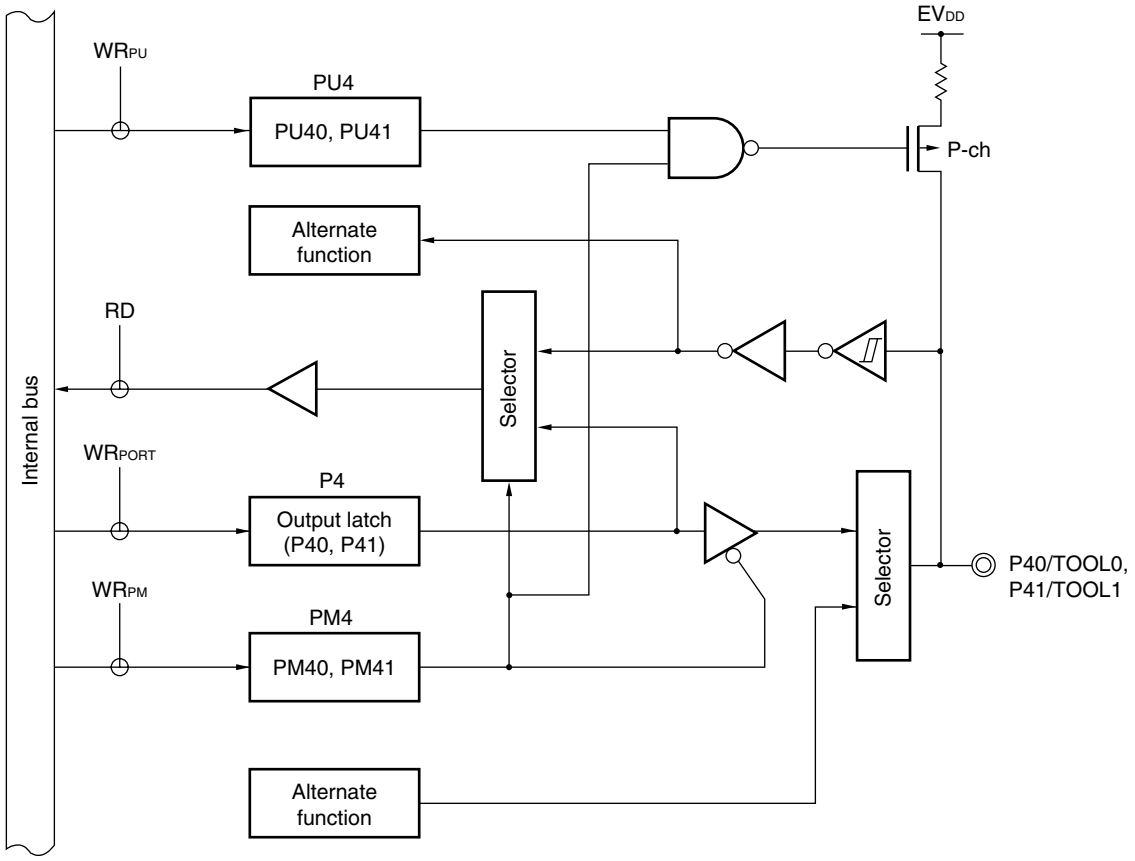
Figure 4-6 shows a block diagram of port 4.

Note When a tool is connected, the P40 and P41 pins cannot be connected to a pull-up resistor.

Caution When a tool is connected, the P40 pin cannot be used as a port pin.
When the on-chip debug function is used, P41 pin can be used as follows by the mode setting on the debugger.

- **1-line mode: can be used as a port (P41).**
- **2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).**

Figure 4-6. Block Diagram of P40, P41



P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

4.2.5 Port 5

78K0R/LG3-M (100 pin: μPD78F8070)
P50/RxD3/SEG39
P51/TxD3/SEG38
P52/TI02/SEG37
P53/TI04/SEG36
P54/SEG35
P55/SEG34
P56/SEG33
P57/SEG32

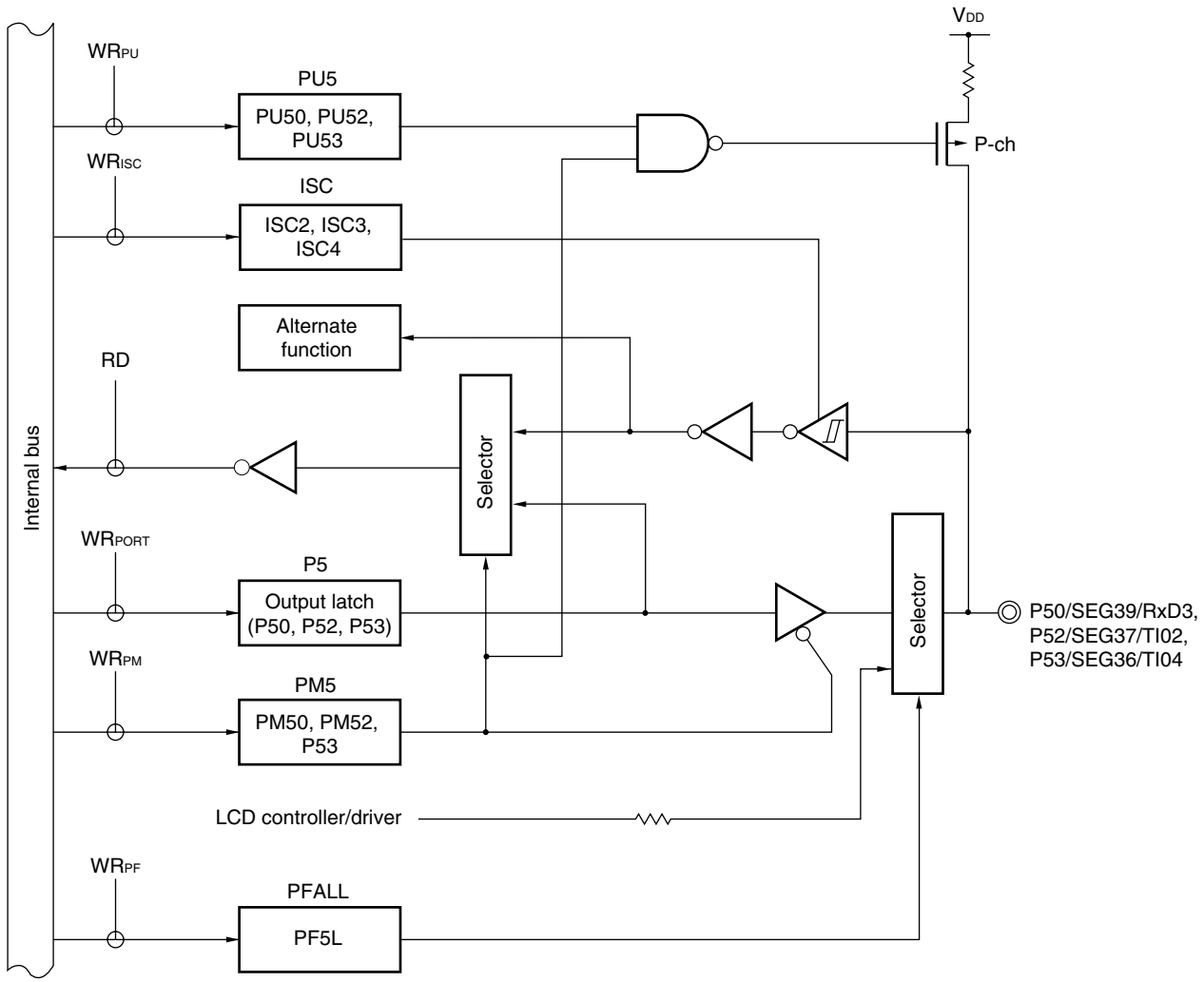
Port 5 is an I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P57 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

This port can also be used for serial interface data I/O, timer input and segment output of LCD controller/driver.

Reset signal generation sets port 5 to input mode.

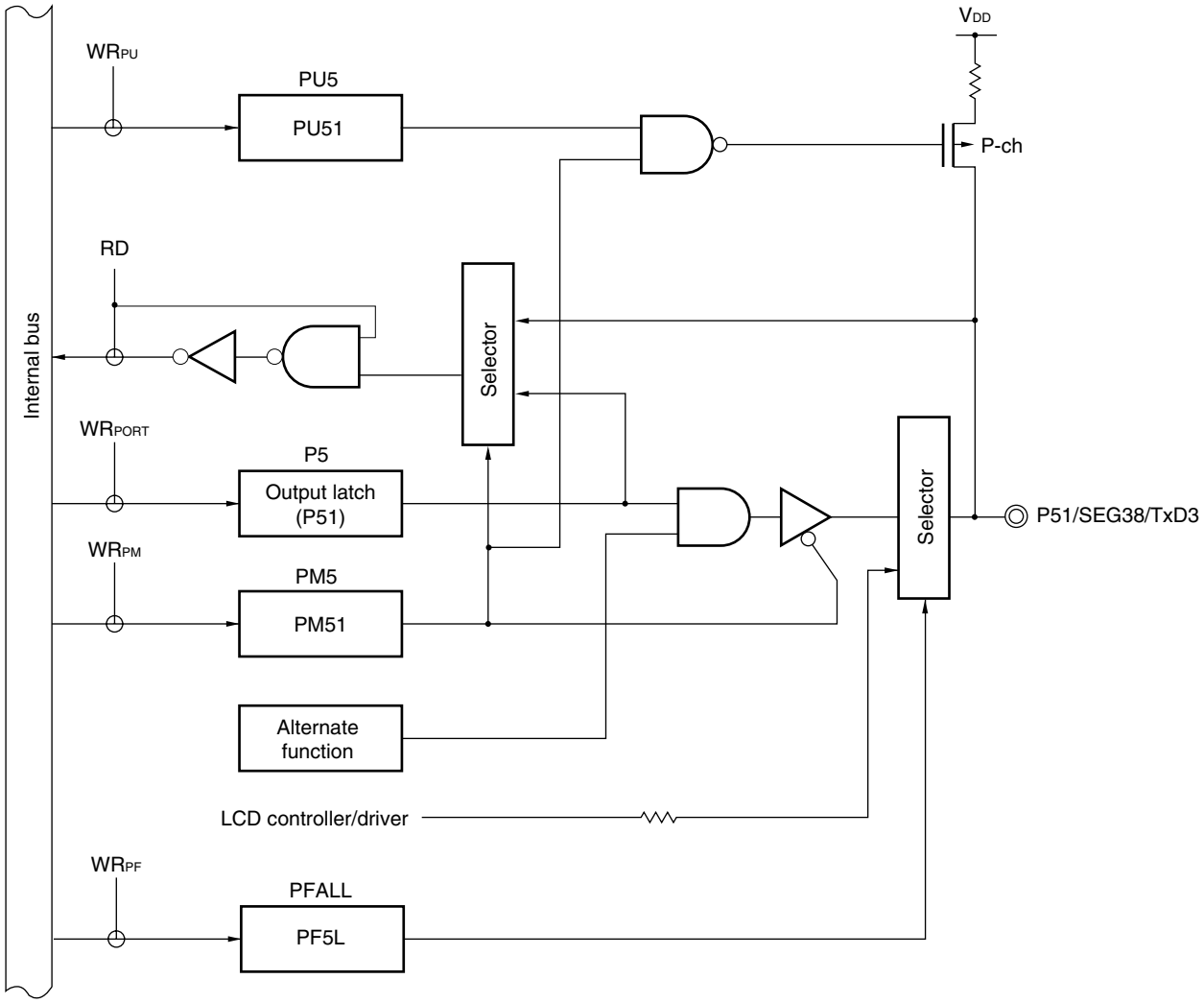
Figures 4-7 to 4-9 show block diagrams of port 5.

Figure 4-7. Block Diagram of P50, P52, and P53



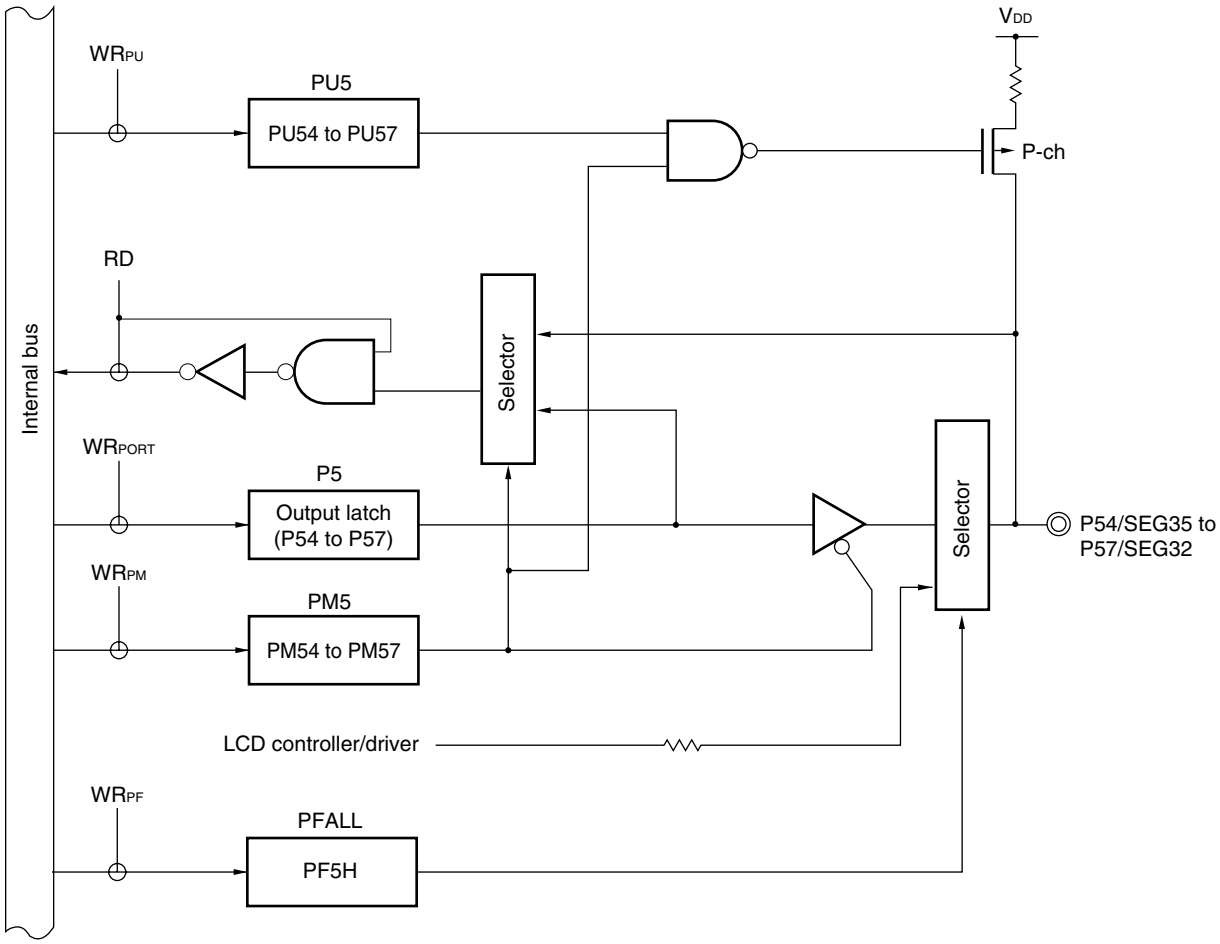
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PFALL: Port function register
ISC: Input switch control register
RD: Read signal
WR××: Write signal

Figure 4-8. Block Diagram of P51



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PFALL: Port function register
RD: Read signal
WR××: Write signal

Figure 4-9. Block Diagram of P54 to P57



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PFALL: Port function register
RD: Read signal
WR××: Write signal

4.2.6 Port 6

78K0R/LG3-M (100 pin: μPD78F8070)
P60/SCL0
P61/SDA0

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 is N-ch open-drain output (6 V tolerance).

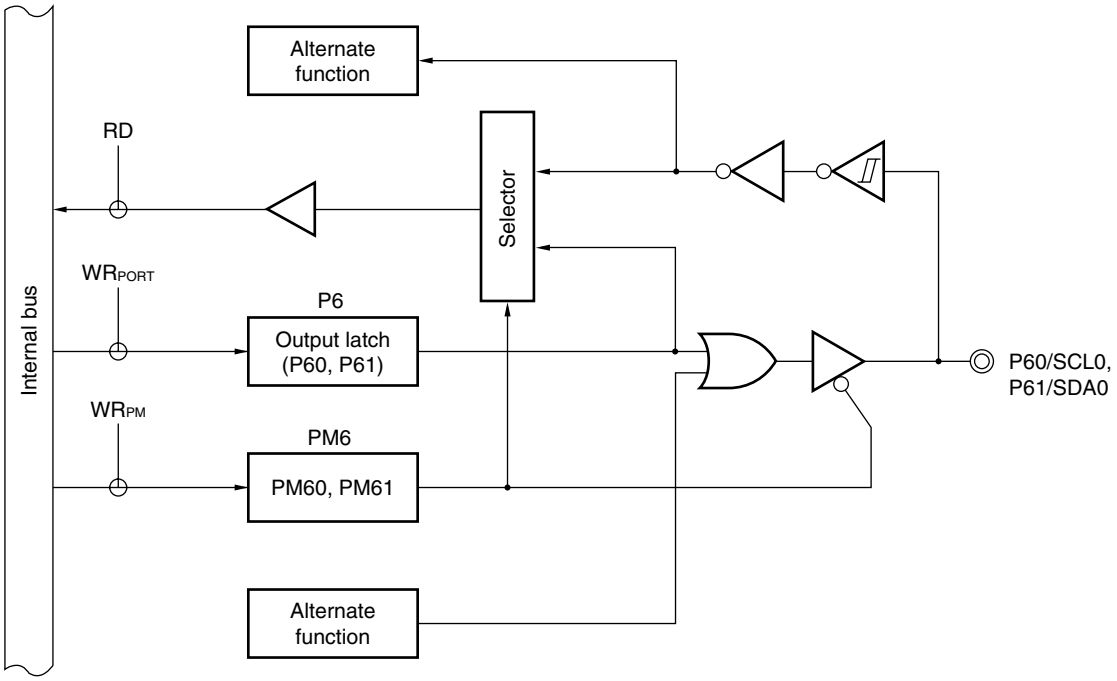
This port can also be used for serial interface IICA data I/O and clock I/O.

Reset signal generation sets port 6 to input mode.

Figure 4-10 shows a block diagram of port 6.

Caution When using P60/SCL0 and P61/SDA0 as a general-purpose port, stop the operation of serial interface IICA.

Figure 4-10. Block Diagram of P60 and P61



P6: Port register 6
PM6: Port mode register 6
RD: Read signal
WR××: Write signal

4.2.7 Port 8

78K0R/LG3-M (100 pin: μPD78F8070)
P81/RxD0/INTP9
P82/TxD0

Port 8 is an I/O port with an output latch. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8). When the P81 and P82 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 8 (PU8).

Output from the P82 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 8 (POM8).

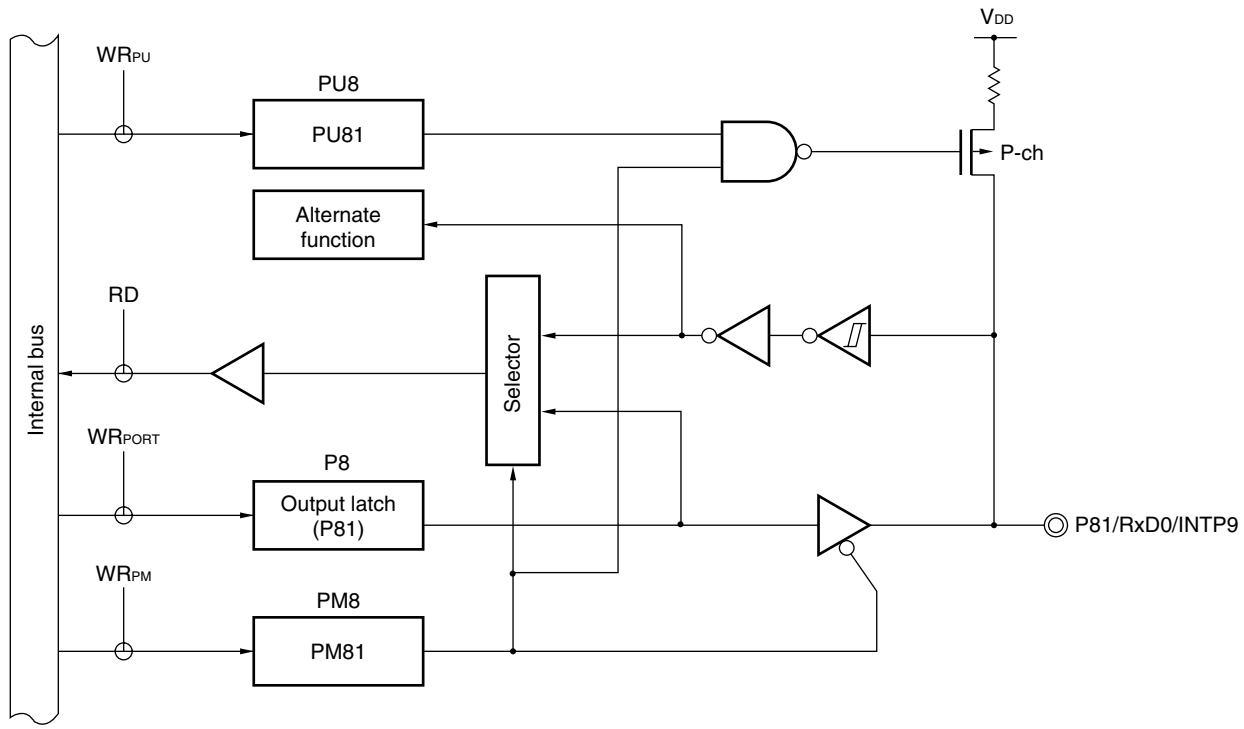
This port can also be used for serial interface clock I/O, data I/O, and external interrupt request input.

Reset signal generation sets port 8 to input mode.

Figures 4-11 and 4-12 show block diagrams of port 8.

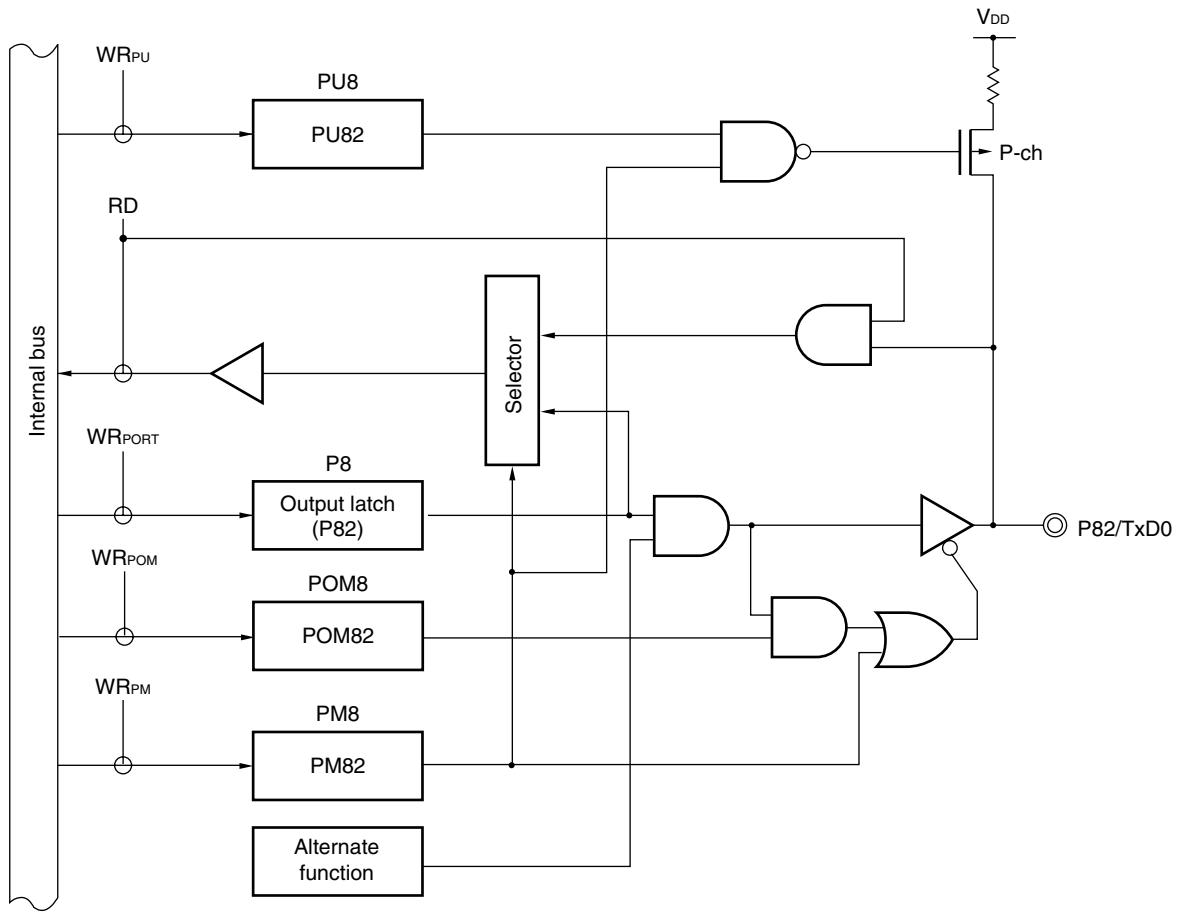
Caution To use P81/RxD0/INTP9, and P82/TxD0, as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 12-5 Relationship Between Register Settings and Pins (Channel 0 of unit 0: UART0 Reception).

Figure 4-11. Block Diagram of P81



P8: Port register 8
PU8: Pull-up resistor option register 8
PM8: Port mode register 8
RD: Read signal
WR××: Write signal

Figure 4-12. Block Diagram of P82



P8: Port register 8
PU8: Pull-up resistor option register 8
POM8: Port output mode register 8
PM8: Port mode register 8
RD: Read signal
WR××: Write signal

4.2.8 Port 9

78K0R/LG3-M (100 pin: μPD78F8070)
P90/SEG31
P91/SEG30
P92/SEG29
P93/SEG28
P94/SEG27
P95/SEG26
P96/SEG25
P97/SEG24

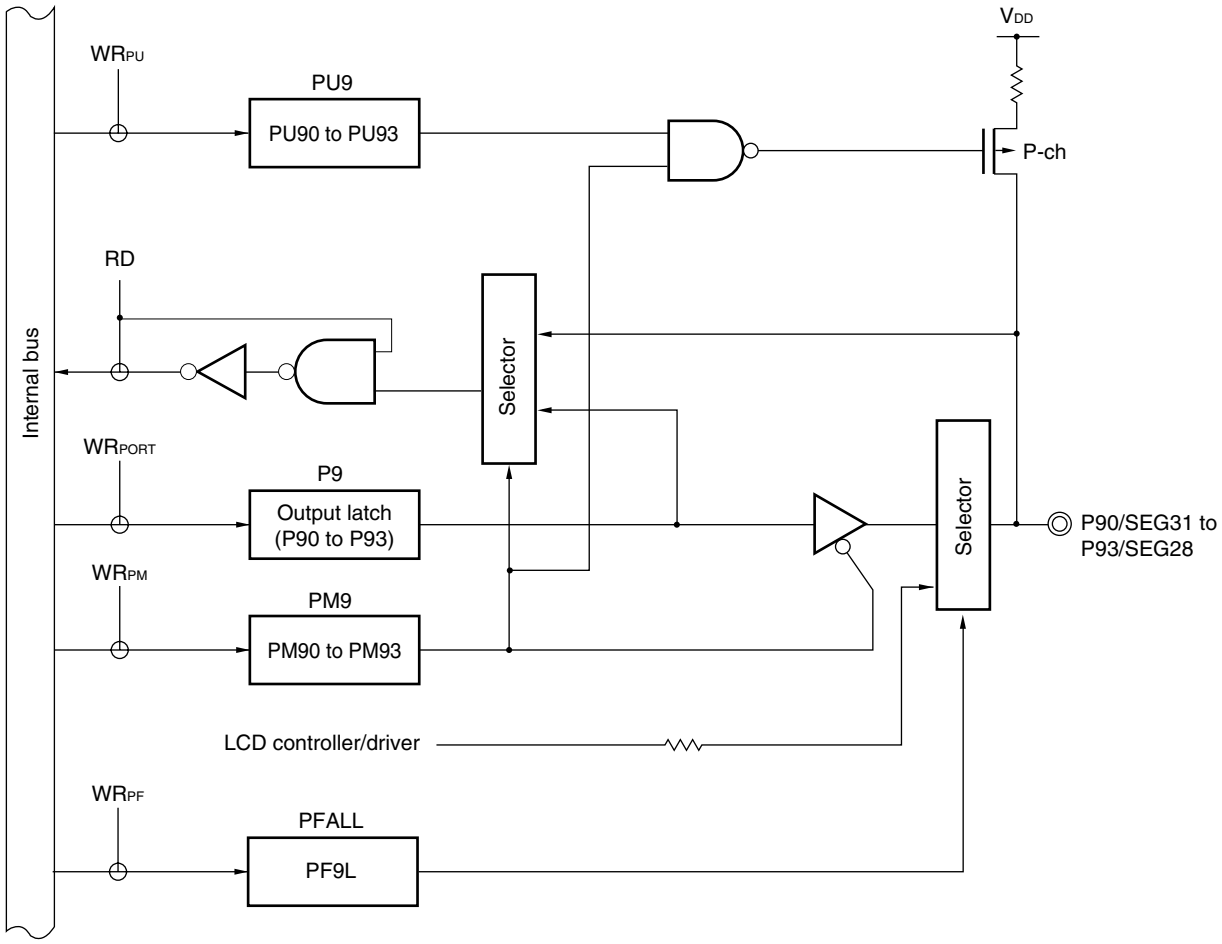
Port 9 is an I/O port with an output latch. Port 9 can be set to the input mode or output mode in 1-bit units using port mode register 9 (PM9). When the P90 to P97 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 9 (PU9).

This port can also be used for segment output.

Reset signal generation sets port 9 to input mode.

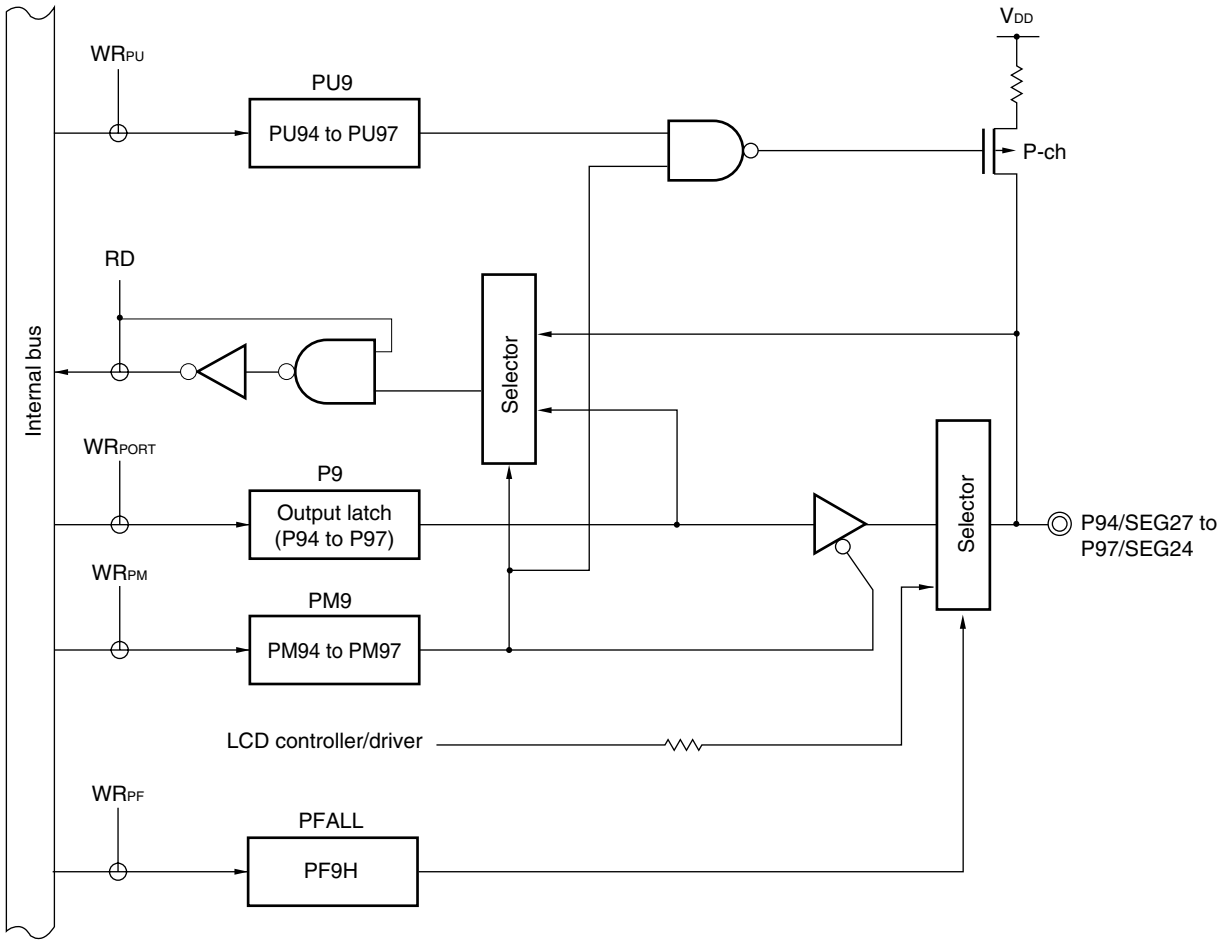
Figures 4-13 and 4-14 show block diagrams of port 9.

Figure 4-13. Block Diagram of P90 to P93



P9: Port register 9
PU9: Pull-up resistor option register 9
PM9: Port mode register 9
PFALL: Port function register
RD: Read signal
WR××: Write signal

Figure 4-14. Block Diagram of P94 to P97



P9: Port register 9
PU9: Pull-up resistor option register 9
PM9: Port mode register 9
PFALL: Port function register
RD: Read signal
WR××: Write signal

4.2.9 Port 10

78K0R/LG3-M (100 pin: μPD78F8070)
P100/SEG15

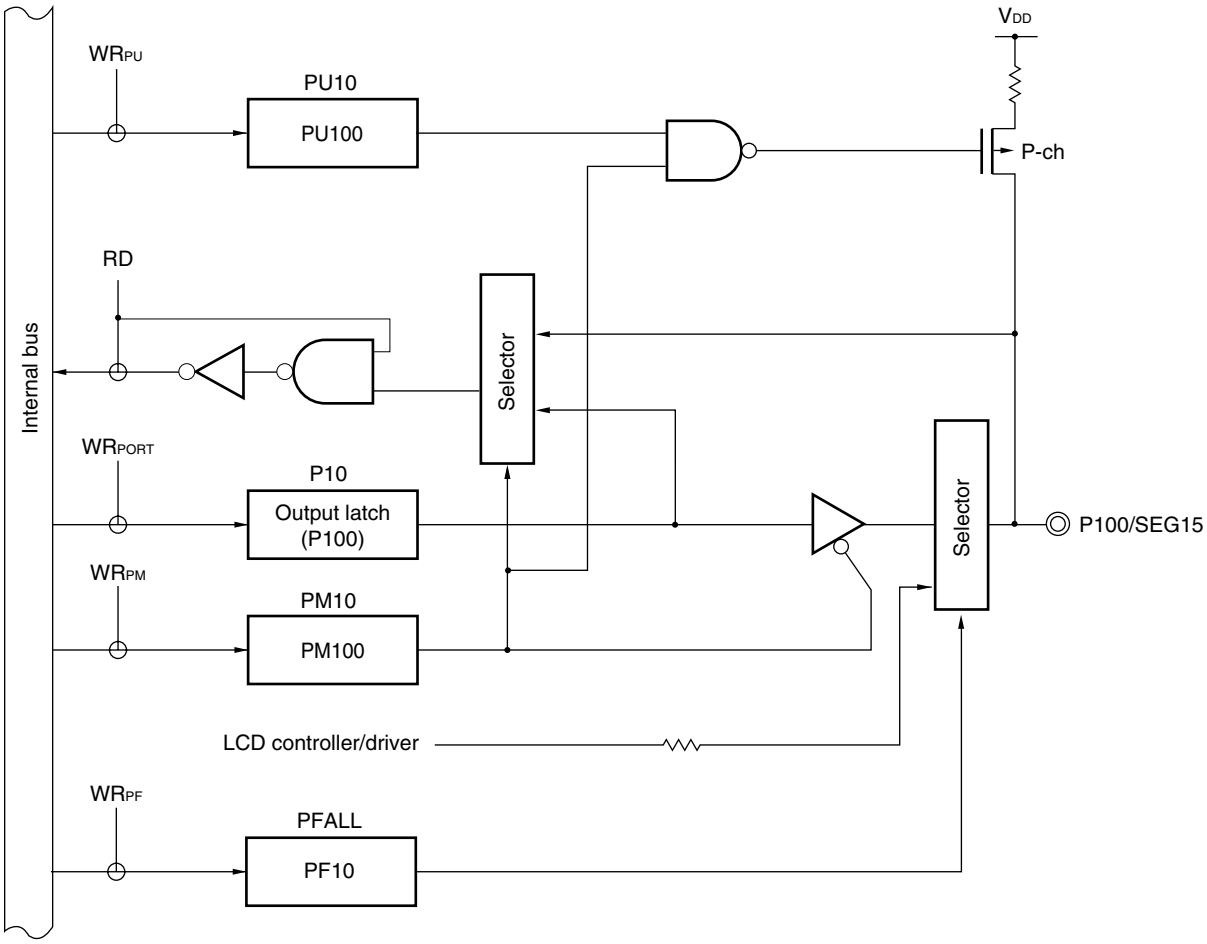
Port 10 is an I/O port with an output latch. Port 10 can be set to the input mode or output mode in 1-bit units using port mode register 10 (PM10). When the P100 pin is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 10 (PU10).

This port can also be used for segment output.

Reset signal generation sets port 10 to input mode.

Figure 4-15 shows a block diagram of port 10.

Figure 4-15. Block Diagram of P100



P10: Port register 10
PU10: Pull-up resistor option register 10
PM10: Port mode register 10
PFALL: Port function register
RD: Read signal
WR××: Write signal

4.2.10 Port 12

78K0R/LG3-M (100 pin: μPD78F8070)
P120/INTP0/EXLVI
P121/X1
P122/X2/EXCLK
P124/EXCLKS0

P120 is a 1-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, P122 and P124 are 3-bit input ports.

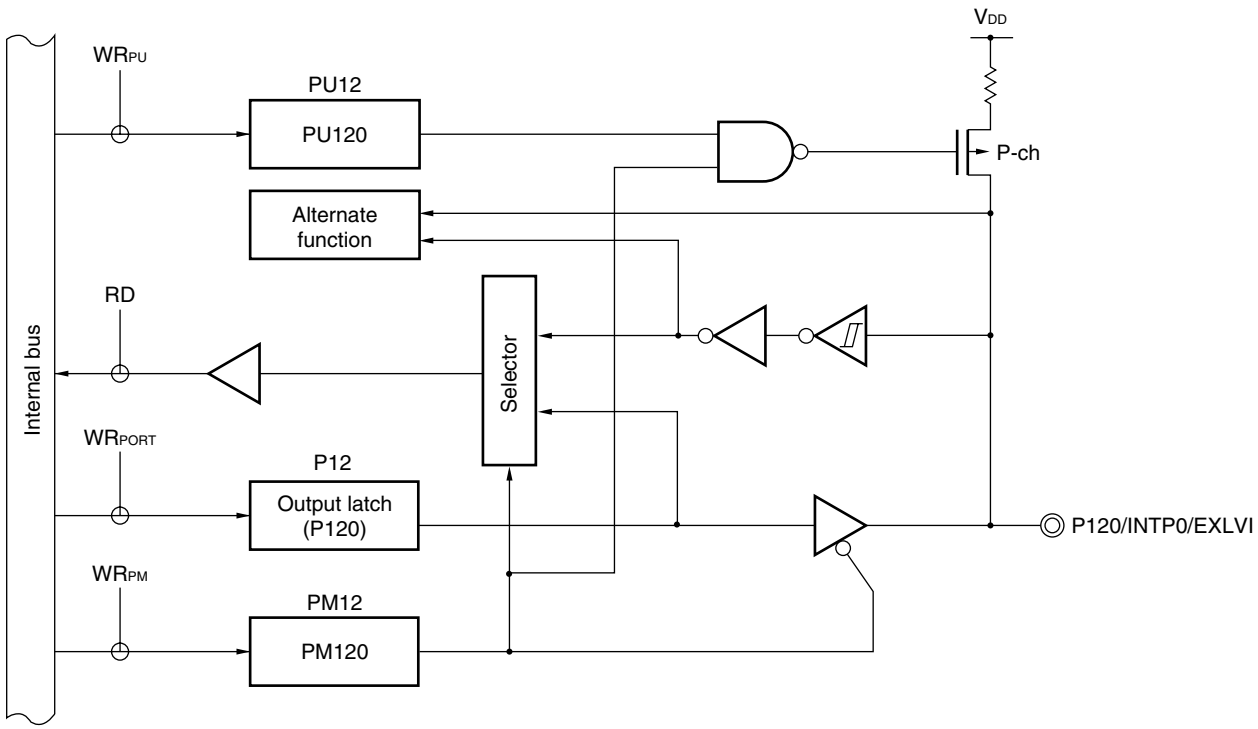
This port can also be used for external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, external clock input for main system clock, and external clock input for subsystem clock.

Reset signal generation sets port 12 to input mode.

Figures 4-16 to 4-18 show block diagrams of port 12.

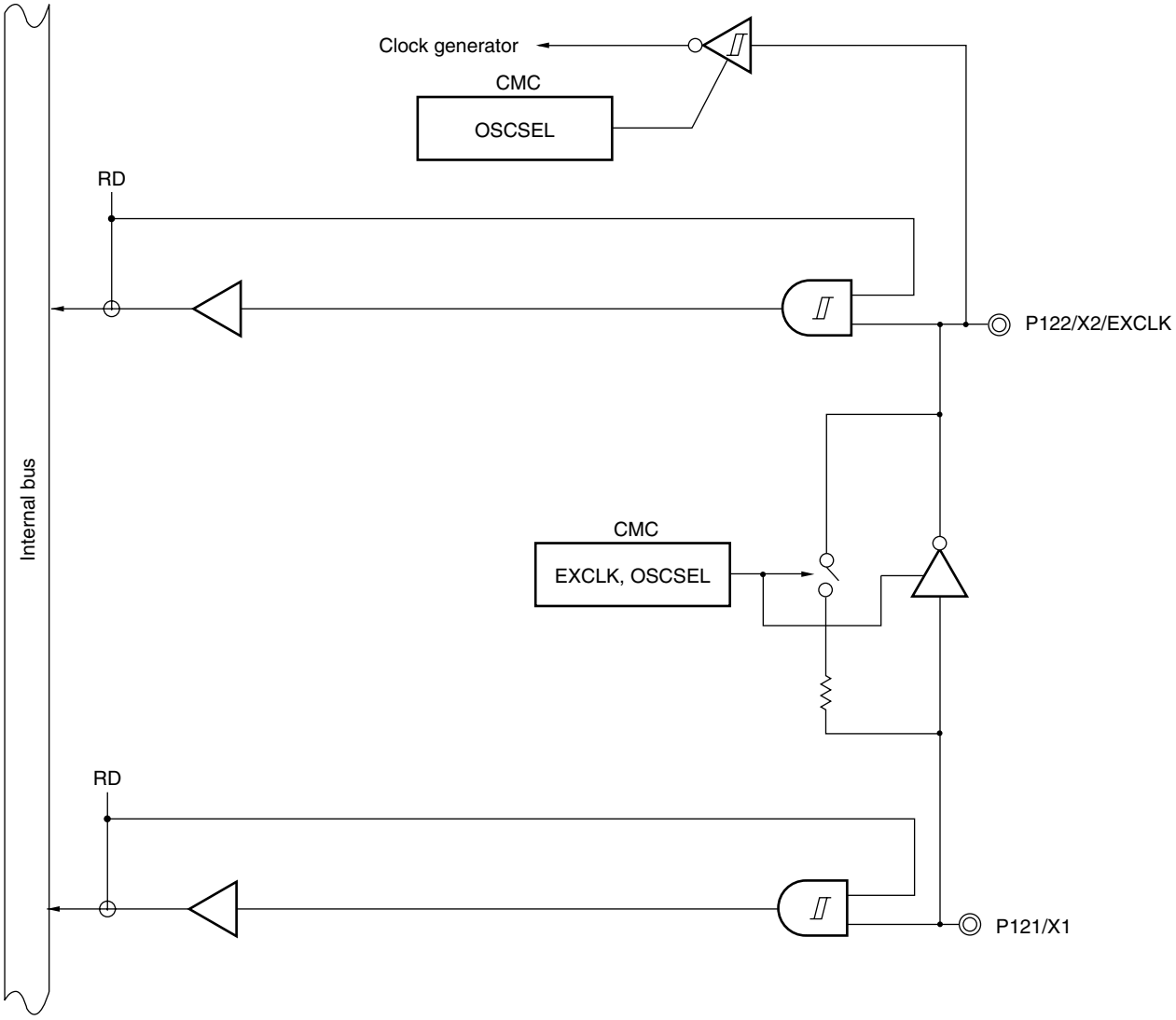
Caution The function setting on P121, P122 and P124 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.

Figure 4-16. Block Diagram of P120



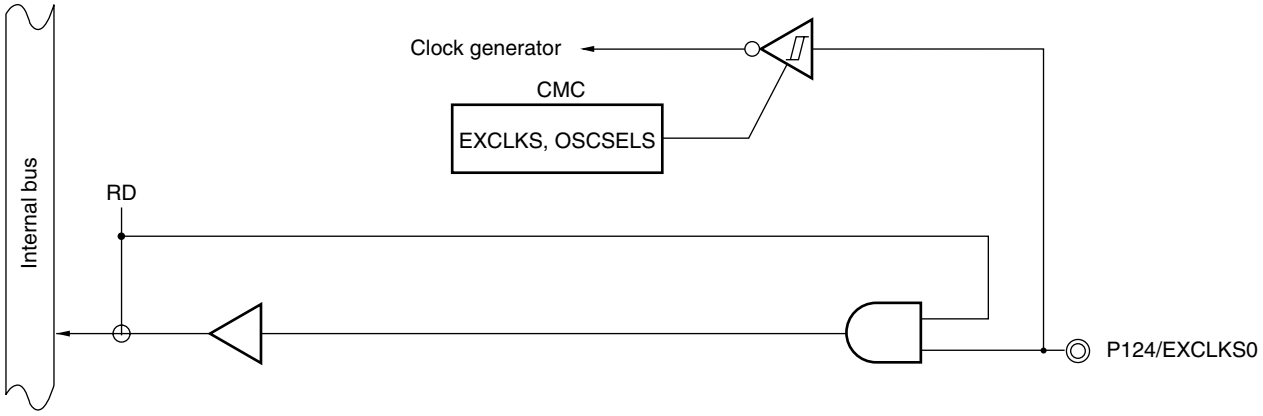
P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
RD: Read signal
WR××: Write signal

Figure 4-17. Block Diagram of P121 and P122



CMC: Clock operation mode control register
RD: Read signal

Figure 4-18. Block Diagram of P124



CMC: Clock operation mode control register
RD: Read signal

4.2.11 Port 14

78K0R/LG3-M (100 pin: μPD78F8070)
P140/SEG23
P141/SEG22
P142/SEG21
P143/SEG20
P144/SEG19
P145/SEG18
P146/SEG17
P147/SEG16

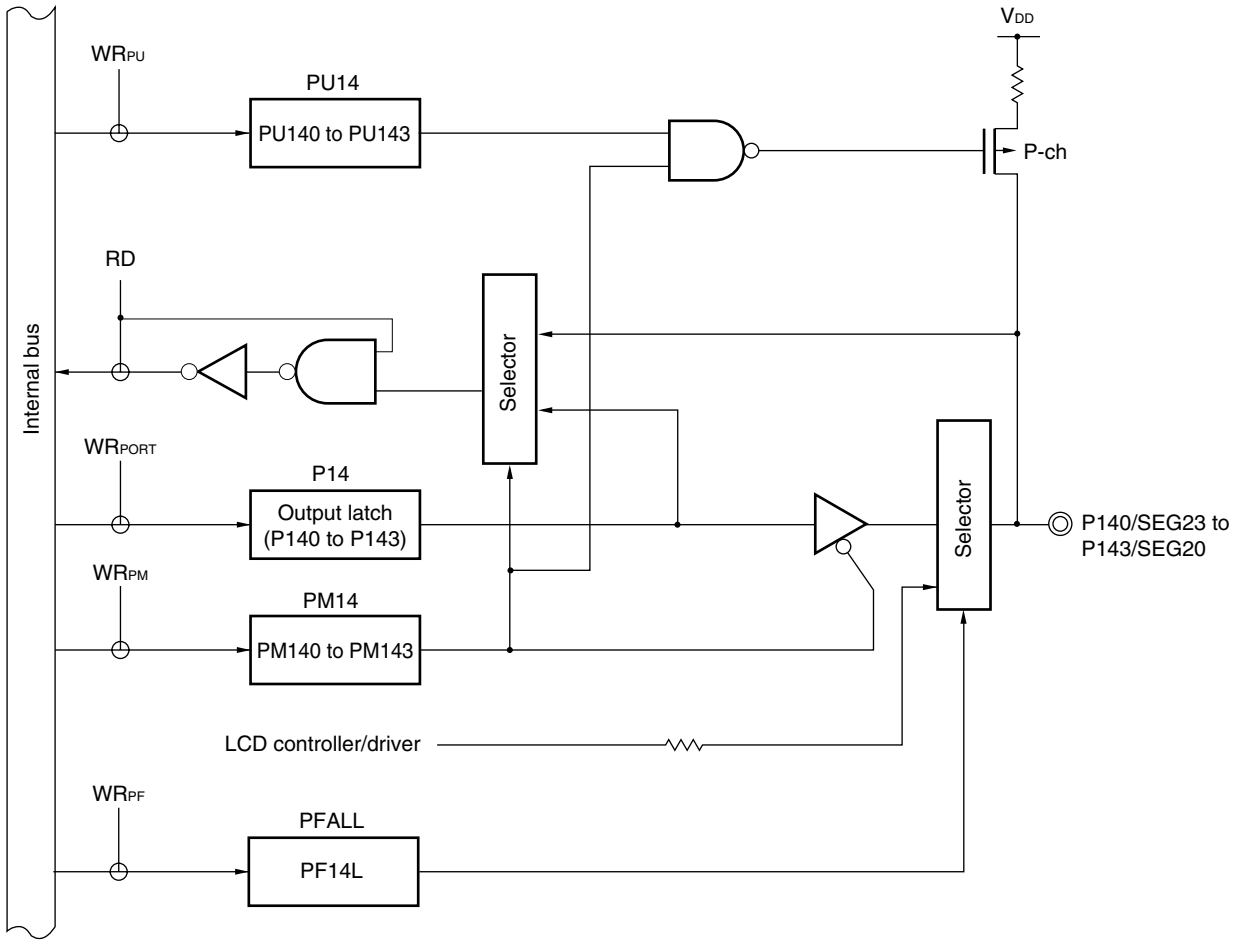
Port 14 is an I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140 to P147 pin is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 14 (PU14).

This port can also be used for segment output.

Reset signal generation sets Port 14 to input mode.

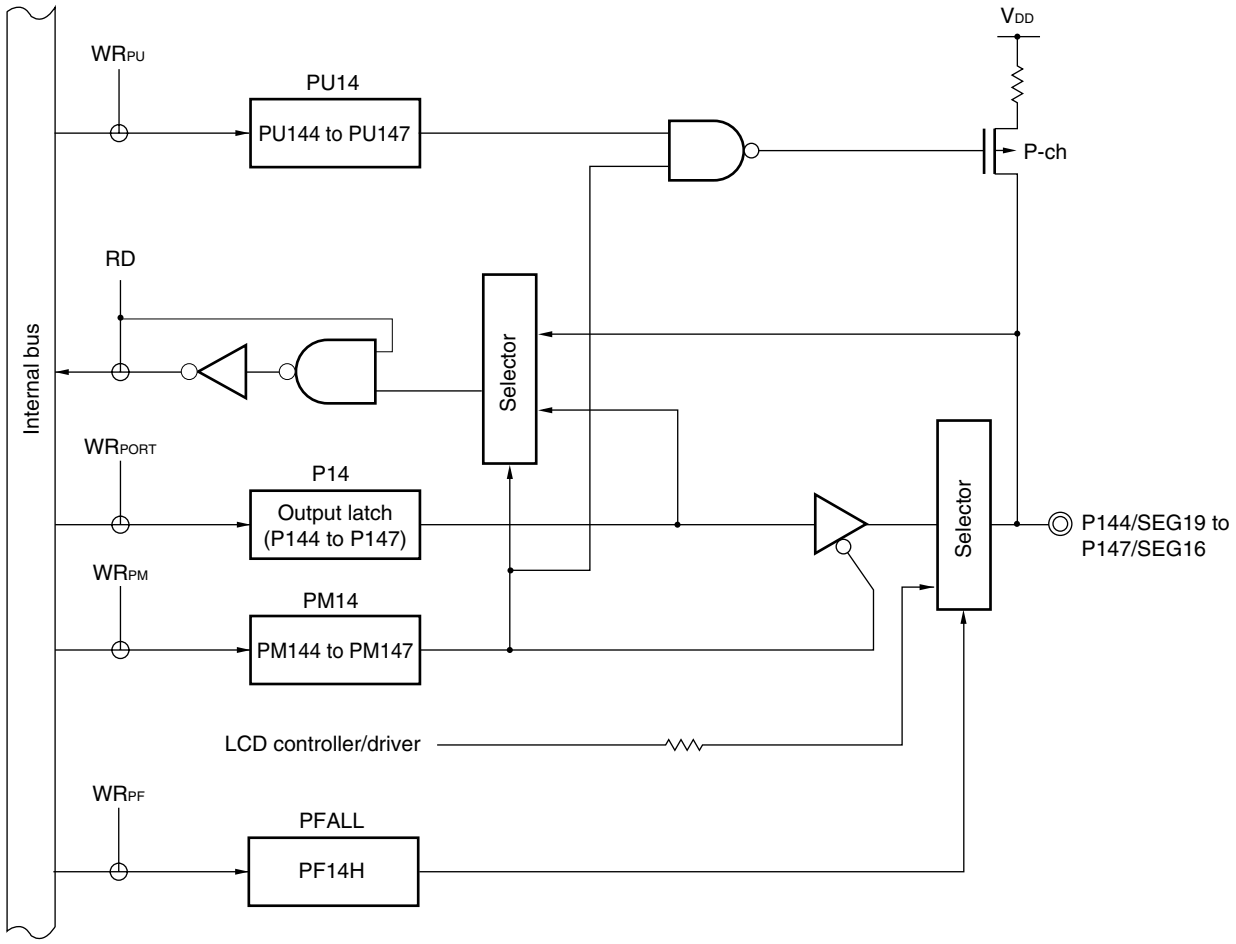
Figures 4-19 and 4-20 show block diagrams of port 14.

Figure 4-19. Block Diagram of P140 to P143



P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
PFALL: Port function register
RD: Read signal
WR××: Write signal

Figure 4-20. Block Diagram of P144 to P147



P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
PFALL: Port function register
RD: Read signal
WR××: Write signal

4.2.12 Port 15

78K0R/LG3-M (100 pin: μPD78F8070)
P152/ANI10
P157/ANI15

Port 15 is an I/O port with an output latch. Port 15 can be set to the input mode or output mode in 1-bit units using port mode register 15 (PM15).

This port can also be used for 10-bit successive approximation type A/D converter analog input.

To use P152/ANI10 and P157/ANI15 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using PM15. Use these pins starting from the lower bit.

To use P152/ANI10 and P157/ANI15 as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM15.

All P152/ANI10 and P157/ANI15 are set in the digital input mode when the reset signal is generated.

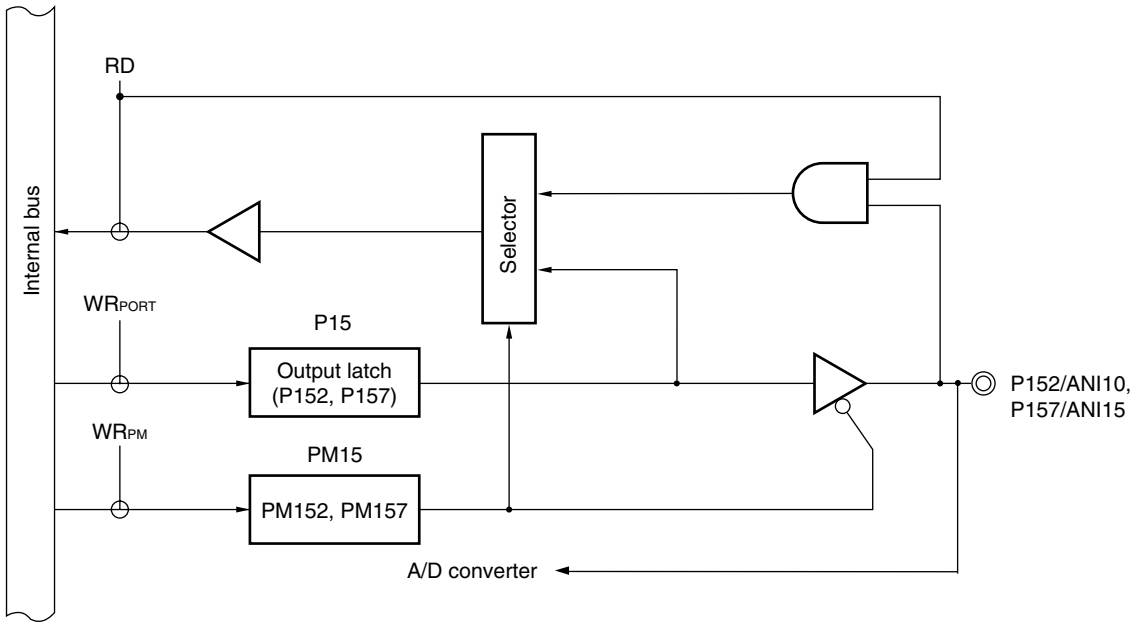
Figures 4-21 show block diagrams of port 15.

Caution Make the AV_{DD} pin the same potential as the V_{DD} pin when port 15 is used as a digital port.

Table 4-4. Setting Functions of ANI10/P152 and ANI15/P157 Pins

ADPC Register	PM15 Register	ADS Register	ANI10/P152 and ANI15/P157 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be A/D converted)
		Does not select ANI.	Analog input (not to be A/D converted)
	Output mode	–	Setting prohibited

Figure 4-21. Block Diagram of P152, P157



P15: Port register 15
PM15: Port mode register 15
RD: Read signal
WR××: Write signal

<R> 4.2.13 Port L0

78K0R/LG3-M (100 pin: μPD78F8070)
LP01/ZX1

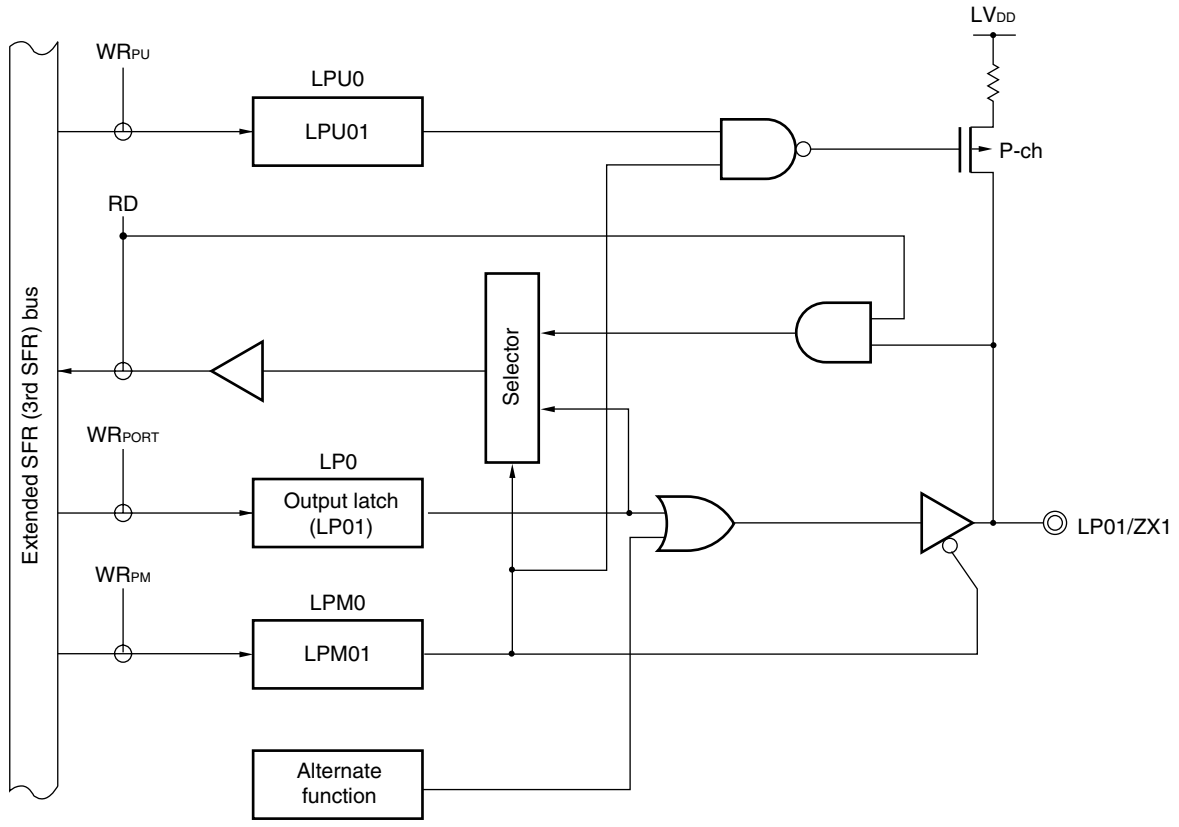
Port L0 is an I/O port with an output latch. Port L0 can be set to the input mode or output mode in 1-bit units using port mode register LP0 (LPM0). When used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register LP0 (LPU0).

This port can also be used for zero-crossing detection signal output of the power quality measurement circuit.

Reset signal generation sets port L0 to input mode.

Figure 4-22 shows a block diagram of port L0.

Figure 4-22. Block Diagram of LP01



LP0: Port register LP0
LPU0: Pull-up resistor option register LP0
LPM0: Port mode register LP0
RD: Read signal
WR××: Write signal

4.3 Registers Controlling Port Function

Port functions are controlled by the following eight types of registers.

- Port mode registers (PMxx, LPM0)
- Port registers (Pxx, LP0)
- Pull-up resistor option registers (Puxx, LPU0)
- Port input mode registers (PIMx)
- Port output mode registers (POMx)
- A/D port configuration register (ADPC)
- Port function register (PFALL)
- Input switch control register (ISC)

(1) Port mode registers (PMxx, LPM0)

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

Port mode register PMxx can be set by a 1-bit or 8-bit memory manipulation instruction.

Port mode register LPM0 is set by using the extended SFR (3rd SFR) interface.

Reset signal generation sets these registers to FFH.

Figure 4-23 Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	PM02 Note1	PM01	PM00	FFF20H	FFH	R/W
PM1	1	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27 Note1	PM26 Note1	PM25 Note1	PM24 Note1	PM23 Note1	PM22 Note1	PM21 Note1	PM20 Note2	FFF22H	FFH	R/W
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	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
PM8	1	1	1	1	1	PM82	PM81	PM80 Note1	FFF28H	FFH	R/W
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FFF29H	FFH	R/W
PM10	1	1	1	1	1	1	1	PM100	FFF2AH	FFH	R/W
PM11	1	1	1	1	1	1	PM111 Note1	PM110 Note1	FFF2BH	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	PM147	PM146	PM145	PM144	PM143	PM142	PM141	PM140	FFF2EH	FFH	R/W
PM15	PM157	1	1	1	1	PM152	PM151 Note1	PM150 Note1	FFF2FH	FFH	R/W
LPM0	1	1	1	1	1	1	LPM01	1	8CH	FFH	R/W

PMmn, LPM01	Pmn, LP01 pin I/O mode selection (m = 0 to 6, 8 to 12, 14, 15 ; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

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

Notes 1. For PM02, PM21 to PM27, PM80, PM110, PM111, PM150, and PM151 bits, be sure to set 0 after reset is released.

2. When using P16/TO05, be sure to set 0 after reset is released.

Caution Be sure to set bits 3 to 7 of PM0, bit 7 of PM1, bits 5 to 7 of PM3, bits 2 to 7 of PM4, bits 2 to 7 of PM6, bits 3 to 7 of PM8, bits 1 to 7 of PM10, bits 2 to 7 of PM11, bits 1 to 7 of PM12, bits 3 to 6 of PM15, and bits 0, 2 to 7 of LPM0 to 1.

(2) Port registers (Pxx, LP0)

These registers write the data that is output from the chip when data is output from a port.

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

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

Port register LP0 is set by using the extended SFR (3rd SFR) interface.

Reset signal generation clears these registers to 00H.

Note It is always 0 and never a pin level that is read out if a port is read during the input mode when P15 is set to function as an analog input for a 10-bit successive approximation type A/D converter.

Figure 4-24. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	0 Note 1	P01	P00	FFF00H	00H (output latch)	R/W
P1	0	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	0 Note 1	0 Note 1	0 Note 1	0 Note 1	0 Note 1	0 Note 1	0 Note 1	P20 Note 2	FFF02H	00H (output latch)	R/W
P3	0	0	0	P34	P33	P32	P31	P30	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	0	0	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
P8	0	0	0	0	0	P82	P81	0 Note 1	FFF08H	00H (output latch)	R/W
P9	P97	P96	P95	P94	P93	P92	P91	P90	FFF09H	00H (output latch)	R/W
P10	0	0	0	0	0	0	0	P100	FFF0AH	00H (output latch)	R/W
P12	0	0	0	P124	0	P122	P121	P120	FFF0CH	Undefined	RW Note 3
P13	0	0	0	0	0	0	0	P130 Note 4	FFF0DH	00H (output latch)	R/W
P14	P147	P146	P145	P144	P143	P142	P141	P140	FFF0EH	00H (output latch)	R/W
P15	P157	0	0	0	0	P152	0 Note 1	0 Note 1	FFF0FH	00H (output latch)	R/W
LP0	0	0	0	0	0	0	LP01	0	8BH	00H (output latch)	R/W

Pmn, LP0	m = 0 to 6, 8 to 10, 12-15 ; n = 0 to 7	
	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. Be sure to set bit 2 of P0, bits 1 to 7 of P2, bit 0 of P8, bits 0 and 1 of P15 to 0 after reset is released.

2. When using P16/TO05, be sure to set 1 after reset is released.

3. P121, P122, and P124 are read-only.

4. P130 is used for the reset of the extended SFR (3rd SFR) (See **CHAPTER 23 RESET FUNCTIONS**.)

(3) Pull-up resistor option registers (Puxx, LPU0)

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 input mode of the pins to which the use of an on-chip pull-up resistor has been specified in these registers.

On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of these registers, PUxx or LPU0.

The pull-up resistor option register PUxx can be set by a 1-bit or 8-bit memory manipulation instruction.

The pull-up resistor option register LPU0 is set by using the extended SFR (3rd SFR) interface.

Reset signal generation clears these registers to 00H.

Figure 4-25. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	0[Note]	PU01	PU00	F0030H	00H	R/W
PU1	0	0[Note]	0[Note]	0[Note]	0[Note]	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0[Note]	PU33	0[Note]	0[Note]	0[Note]	F0033H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	00H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU8	0	0	0	0	0	PU82	PU81	0[Note]	F0038H	00H	R/W
PU9	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90	F0039H	00H	R/W
PU10	0	0	0	0	0	0	0	PU100	F003AH	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	PU147	PU146	PU145	PU144	PU143	PU142	PU141	PU140	F003EH	00H	R/W
LPU0	0	0	0	0	0	0	LPU01	0	8DH	00H	R/W

PUmn, LPU0	Pmn, LPU0 pin on-chip pull-up resistor selection (m = 0, 1, 3 to 5, 8 to 10, 12, 14 ; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Note Be sure to set bit 2 of PU0, bits 3 to 6 of PU1, bits 0 to 2, and 4 of PU3, bit 0 of PU8 to 0.

(4) Port input mode register (PIMx)

This register sets the input buffer of P10 and P11 in 1-bit units.

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

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

Reset signal generation clears these registers to 00H.

Figure 4-26. Format of Port Input Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM1	0	0[Note]	0[Note]	0[Note]	0[Note]	0	PIM11	PIM10	F0041H	00H	R/W

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

Note Be sure to set bits 3 to 6 to 0.

(5) Port output mode registers (POMx)

These registers set the output mode of P10 to P12, and P82 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 SDA20 pin during simplified I^2C communication with an external device of the same potential.

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

Reset signal generation clears these registers to 00H.

Figure 4-27. Format of Port Output Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM1	0	0 Note	0 Note	0 Note	0 Note	POM12	POM11	POM10	F0051H	00H	R/W
POM8	0	0	0	0	0	POM82	0	0	F0058H	00H	R/W

POMmn	Pmn pin output mode selection (m = 1, 8 ; n = 0 to 2)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

Note Be sure to set bits 3 to 6 of POM1 to 0.

(6) A/D port configuration register (ADPC)

This register switches the ANI10/P152 and ANI15/P157 pins to analog input or digital I/O of port.

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

Reset signal generation sets this register to 10H.

Figure 4-28. Format of A/D Port Configuration Register (ADPC)

Address: F0017H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching	
					Port 15	
					ANI15/P157	ANI10/P152
0	1	0	1	0	A	A
0	1	1	1	1	A	D
1	0	0	0	0	D	D
Other than the above					Setting prohibited	

Cautions 1. Set a channel to be used for A/D conversion in the input mode by using port mode register 15 (PM15).

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

(7) Port function register (PFALL)

This register sets whether to use pins P50 to P57, P90 to P97, P100, and P140 to P147 as port pins (other than segment output pins) or segment output pins.

PFALL is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PFALL to 00H.

Figure 4-29. Format of Port Function Register (PFALL)

Address: F0080H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFALL	0	PF14H	PF14L	PF10	PF9H	PF9L	PF5H	PF5L

PF14H	Port/segment outputs specification of the P144 to P147 pins
0	Used the P144 to P147 pins as port (other than segment output)
1	Used the P144 to P147 pins as segment output

PF14L	Port/segment outputs specification of the P140 to P143 pins
0	Used the P140 to P143 pins as port (other than segment output)
1	Used the P140 to P143 pins as segment output

PF10	Port/segment outputs specification of the P100 pin
0	Used the P100 pin as port (other than segment output)
1	Used the P100 pin as segment output

PF9H	Port/segment outputs specification of the P94 to P97 pins
0	Used the P94 to P97 pins as port (other than segment output)
1	Used the P94 to P97 pins as segment output

PF9L	Port/segment outputs specification of P90 to P93 pins
0	Used the P90 to P93 pins as port (other than segment output)
1	Used the P90 to P93 pins as segment output

PF5H	Port/segment outputs specification of the P54 to P57 pins
0	Used the P54 to P57 pins as port (other than segment output)
1	Used the P54 to P57 pins as segment output

PF5L	Port/segment outputs specification of P50 to P53 pins
0	Used the P50 to P53 pins as port (other than segment output)
1	Used the P50 to P53 pins as segment output

Caution Be sure to clear bit 7 to “0”.

(8) Input switch control register (ISC)

Bits 0 and 1 of ISC are used for linking with an external interrupt or a timer array unit when performing a LIN-bus communication operation with UART3.

When bit 0 is set to 1, the input signal of the serial data input (RxD3) 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 (RxD3) pin is selected as a timer input, so that the pulse widths of a sync break field and a sync field can be measured by the timer.

Bits 2 to 4 of ISC are used to prevent through current from entering when using the TI04/SEG36/P53, TI02/SEG37/P52, and RxD3/SEG38/P50 pins as segment outputs or port outputs.

The segment output pins to be used alternatively with the TI04, TI02, and RxD3 pins are internally connected with a Schmitt trigger buffer. When using these pins as segment outputs or port outputs, bits 2 to 4 of ISC must be set to 0 (prohibiting input to Schmitt trigger buffers) in order to prevent through current from entering.

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

Reset signal generation clears this register to 00H.

Figure 4-30. Format of Input Switch Control Register (ISC)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	ISC4	ISC3	ISC2	ISC1	ISC0

ISC4	TI04/SEG36/P53 schmitt trigger buffer control
0	Disables input
1	Enables input

ISC3	TI02/SEG37/P52 schmitt trigger buffer control
0	Disables input
1	Enables input

ISC2	RxD3/SEG39/P50 schmitt trigger buffer control
0	Disables input
1	Enables input

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 RxD3 pin is used as timer input (wakeup signal detection).

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 RxD3 pin as an external interrupt (to measure the pulse widths of the sync break field and sync field).

Caution Be sure to clear bits 5 to 7 to "0".

To use the TI04/SEG36/P53, TI02/SEG37/P52, and RxD3/SEG39/P50 pins, set the PF5L and ISCn (n = 2 to 4) bits as follows, according to the function to be used.

PF5L	ISCn	Pin function
0	0	Port output (default)
0	1	Port input, timer input, or serial data input
1	0	Segment output
1	1	Setting prohibited

4.4 Port Function Operations

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

4.4.1 Writing to I/O port

(1) Output mode

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

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

4.4.2 Reading from I/O port

(1) Output mode

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

(2) Input mode

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

4.4.3 Operations on I/O port

(1) Output mode

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

(2) Input mode

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

4.4.4 Connecting to external device with different power potential (2.5 V)

When parts of ports 1, and 8 operate with V_{DD} = 2.7 V to 3.6 V, I/O connections with an external device that operates on a 2.5V power supply voltage are possible.

Regarding inputs, CMOS/TTL switching is possible on a bit-by-bit basis by port input mode register (PIM1).

Moreover, regarding outputs, different power potentials can be supported by switching the output buffer to the N-ch open drain (V_{DD} tolerance) by the port output mode registers (POM1, and POM8).

(1) Setting procedure when using I/O pins of UART2 and CSI20 functions

(a) Use as 2.5V input port

<1> After reset release, the port mode is the input mode (Hi-Z).

<2> If pull-up is needed, externally pull up the pin to be used (on-chip pull-up resistor cannot be used).

In case of UART2: P11
In case of CSI20: P10, P11

<3> Set the corresponding bit of the PIM1 register to 1 to switch to the TTL input buffer.

<4> V_{IH}/V_{IL} operates on a 2.5V operating voltage.

(b) Use as 2.5V output port

<1> After reset release, the port mode is the input mode (Hi-Z).

<2> Pull up externally the pin to be used (on-chip pull-up resistor cannot be used).

In case of UART2: P12
In case of CSI20: P10, P12

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

<4> Set the corresponding bit of the POM1 register to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.

<5> Set the output mode by manipulating the PM1 register.
At this time, the output data is high level, so the pin is in the Hi-Z state.

<6> Operation is done only in the low level according to the operating status of the serial array unit.

(2) Setting procedure when using I/O pins of simplified IIC20 functions

<1> After reset release, the port mode is the input mode (Hi-Z).

<2> Externally pull up the pin to be used (on-chip pull-up resistor cannot be used).

In case of simplified IIC20: P11, P10

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

<4> Set the corresponding bit of the POM1 register to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.

<5> Set the corresponding bit of the PM1 register 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.

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

4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function

To use the alternate function of a port pin, set the port mode register and output latch as shown in Table 4-5.

Table 4-5. Settings of Port Mode Register and Output Latch When Using Alternate Function (1/3)

Pin Name	Alternate Function		PFALL (PFxxx)	ISC (ISCx)	PMxx	Pxx
	Function Name	I/O				
<R> P00	CAPH	Output	–	–	1	×
<R> P01	CAPL	Output	–	–	1	×
P10	$\overline{\text{SCK20}}$	Input	–	–	1	×
		Output	–	–	0	1
	SCL20	I/O	–	–	0	1
P11	SI20	Input	–	–	1	×
	RxD2	Input	–	–	1	×
	SDA20	I/O	–	–	0	1
	INTP6	Input	–	–	1	×
P12	SO20	Output	–	–	0	1
	TxD2	Output	–	–	0	1
	TO02	Output	–	–	0	0
P16	TO05	Output	–	–	0	0
P33	TI07	Input	–	ISC1 = 0	1	×
	TO07	Output	–	–	0	0
	INTP3	Input	–	–	1	×
P40	TOOL0	I/O	–	–	×	×
P41	TOOL1	Output	–	–	×	×
P50 Note 1	RxD3	Input	PF5L = 0	ISC2 = 1 Note 2	1	×
P51 Note 1	TxD3	Output		–	0	1
P52 Note 1	TI02	Input		ISC3 = 1	1	×
P53 Note 1	TI04	Input		ISC4 = 1	1	×
P60	SCL0	I/O	–	–	0	0
P61	SDA0	I/O	–	–	0	0
P81	RxD0	Input	–	–	1	×
	INTP9	Input	–	–	1	×
P82	TxD0	Output	–	–	0	1

Notes 1. Refer to **Table 4-5 Settings of Port Mode Register and Output Latch When Using Alternate Function (3/3)** about the segment output (SEGxx).

2. The RxD3 input can be set as the input source of an external interrupt input (INTP0) by setting ISC0 = 1.
The RxD3 input can be set as the input source of a timer input (TI07) by setting ISC1 = 1.

Remark ×: don't care
–: Not applicable
PFALL: Port function register
ISC: Input switch control register
PM××: Port mode register
P××: Port output latch

Table 4-5. Settings of Port Mode Register and Output Latch When Using Alternate Function (2/3)

Pin Name	Alternate Function		PFALL (PFxxx)	ISC (ISCx)	PMxx	Pxx
	Function Name	I/O				
P120	INTP0[Note 1]	Input	–	ISC0 = 0	1	×
	EXLVI[Note 1]	Input	–	–	1	×
P121	X1[Note 1]	–	–	–	×	×
P122	X2[Note 1]	–	–	–	×	×
	EXCLK[Note 1]	Input	–	–	×	×
P124	EXCLKS0[Note 2]	Input	–	–	×	×
P152[Note 3]	ANI10	Input	–	–	1	×
P157[Note 3]	ANI15	Input	–	–	1	×
LP01	ZX1[Note 4]	Output	–	–	0	0

Notes 1. To use the P121 and P122 pins for main system clock resonator connection (X1, X2), or main system clock external clock input (EXCLK), the X1 oscillation mode, or external clock input mode must be set, respectively, by using the clock operation mode control register (CMC). CMC can be written only once after reset release (for details, refer to **5.3 (1) Clock operation mode control register (CMC)**). The reset value of CMC is 00H (both P121 and P122 are input port pins).

2. To use the P124 pin for an external clock for the subsystem clock (EXCLKS0), the external clock must be set by using the clock operation mode control register (CMC) and port function control register (PORTCTL). (For details, see **5.3 (1) Clock operation mode control register (CMC)** and **5.3 (2) Port function control register (PORTCTL).**).

3. The functions of the P152/ANI10 and P157/ANI15 pins are determined according to the settings of A/D port configuration register (ADPC), port mode register 15 (PM15), and analog input channel specification register (ADS). For details, see **4.2.12 Port 15**.

4. When using the LP01 pin as the zero-crossing signal ZX1 output, set the zero-crossing output control bit (ZX1EN) of the fault detection control register (PQMCTL) to 1.

Remark ×: don't care
–: Not applicable
PFALL: Port function register
ISC: Input switch control register
PM××: Port mode register
P××: Port output latch

Table 4-5. Settings of Port Mode Register and Output Latch When Using Alternate Function (3/3)

Pin Name	Alternate Function		PFALL (PFxxx)	ISC (ISCx)	PMxx	Pxx
	Function Name	I/O				
P50[Note]	SEG39	Output	PF5L = 1	ISC2 = 0	×	×
P51[Note]	SEG38	Output	PF5L = 1	–	×	×
P52[Note]	SEG37	Output	PF5L = 1	ISC3 = 0	×	×
P53[Note]	SEG36	Output	PF5L = 1	ISC4 = 0	×	×
P54 to 57	SEG35 to SEG32	Output	PF5H = 1	–	×	×
P90 to 93	SEG31 to SEG28	Output	PF9L = 1	–	×	×
P94 to 97	SEG27 to SEG24	Output	PF9H = 1	–	×	×
P140 to 143	SEG23 to SEG20	Output	PF14L = 1	–	×	×
P144 to 147	SEG19 to SEG16	Output	PF14H = 1	–	×	×
P100	SEG15	Output	PF10 = 1	–	×	×

Note Refer to **Table 4-5 Settings of Port Mode Register and Output Latch When Using Alternate Function (1/3)** about the alternative functions other than the segment output (SEGxx).

Remark ×: don't care
–: Not applicable
PFALL: Port function register
ISC: Input switch control register
PM××: Port mode register
P××: Port output latch

4.6 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 P90 is an output port, P91 to P97 are input ports (all pin statuses are high level), and the output latch value of port 9 is 00H, if the output of output port P90 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 78K0R/Lx3-M Microcontrollers.

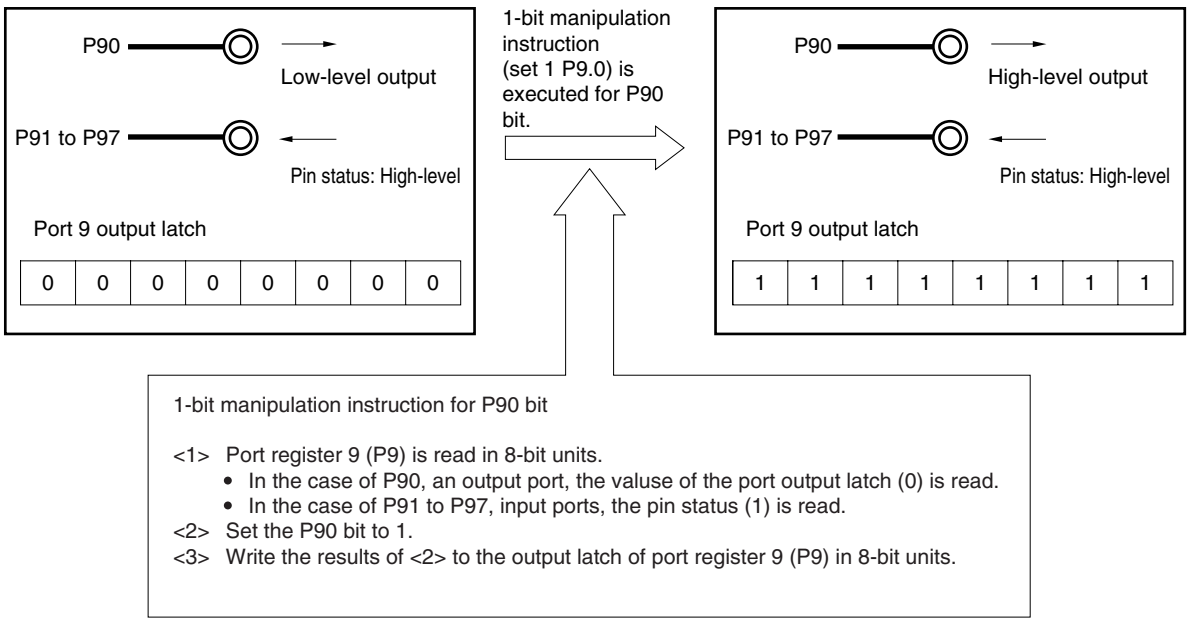
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P90, which is an output port, is read, while the pin statuses of P91 to P97, which are input ports, are read. If the pin statuses of P91 to P97 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 4-31. Bit Manipulation Instruction (P90)



CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

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

(1) Main system clock

<1> X1 oscillator

This circuit oscillates a clock of f_X = 2 to 20 MHz by connecting a resonator to X1 and X2.
Oscillation can be stopped by executing the STOP instruction or setting of MSTOP (bit 7 of the clock operation status control register (CSC)).

<2> Internal high-speed oscillator[Note]

This circuit oscillates clocks of f_{IH} = 1 MHz (TYP.) or f_{IH} = 8 MHz (TYP.). After a reset release, the CPU always starts operating with this internal high-speed oscillation clock. Oscillation can be stopped by executing the STOP instruction or setting HIOSTOP (bit 0 of CSC).

<3> 20 MHz internal high-speed oscillation clock oscillator[Note]

This circuit oscillates a clock of f_{IH20} = 20 MHz (TYP.). Oscillation can be started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1 with $V_{DD} \geq 2.7$ V. Oscillation can be stopped by setting DSCON to 0.

Note To use the internal high-speed oscillation clock, use the option byte to set the frequency (1 MHz, 8 MHz, or 20 MHz) in advance (for details, see **CHAPTER 27 OPTION BYTE**). Also, the internal high-speed oscillator automatically starts oscillating after reset release. To use the 20 MHz internal high-speed oscillator to operate the microcontroller, oscillation is started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1.

<R> **Caution A 10 MHz clock is necessary to operate the ΔΣ-type A/D converter. Supply a 10 MHz or 20 MHz resonator or external clock to the X1 and X2 pins.**

An external main system clock (f_{EX} = 2 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of MSTOP.
As the main system clock, a high-speed system clock (X1 clock or external main system clock) or internal high-speed oscillation clock can be selected by setting of MCM0 (bit 4 of the system clock control register (CKC)).

Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{IH20}: 20 MHz internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency

(2) Subsystem clock

- **XT1 clock oscillator**

  This circuit oscillates a clock of f_{SUB} = 32.768 kHz by connecting a 32.768 kHz resonator to XT1 and XT2.

  An external subsystem clock (f_{EXS} = 32.768 KHz) can also be supplied from the EXCLKS1/XT1 pin.

(3) Internal low-speed oscillation clock (clock for watchdog timer)

- **Internal low-speed oscillator**

  This circuit oscillates a clock of f_{IL} = 30 kHz (TYP.).

  The internal low-speed oscillation clock cannot be used as the CPU clock. The only hardware that operates with the internal low-speed oscillation clock is the watchdog timer.

  Oscillation is stopped when the watchdog timer stops.

Remarks 1. f_{SUB}: Subsystem clock frequency

f_{EXS}: External subsystem clock frequency

f_{IL}: Internal low-speed oscillation clock frequency

2. The watchdog timer stops in the following cases.

- When bit 4 (WDTON) of an option byte (000C0H) = 0
- If the HALT or STOP instruction is executed when bit 4 (WDTON) of an option byte (000C0H) = 1 and bit 0 (WDSTBYON) = 0

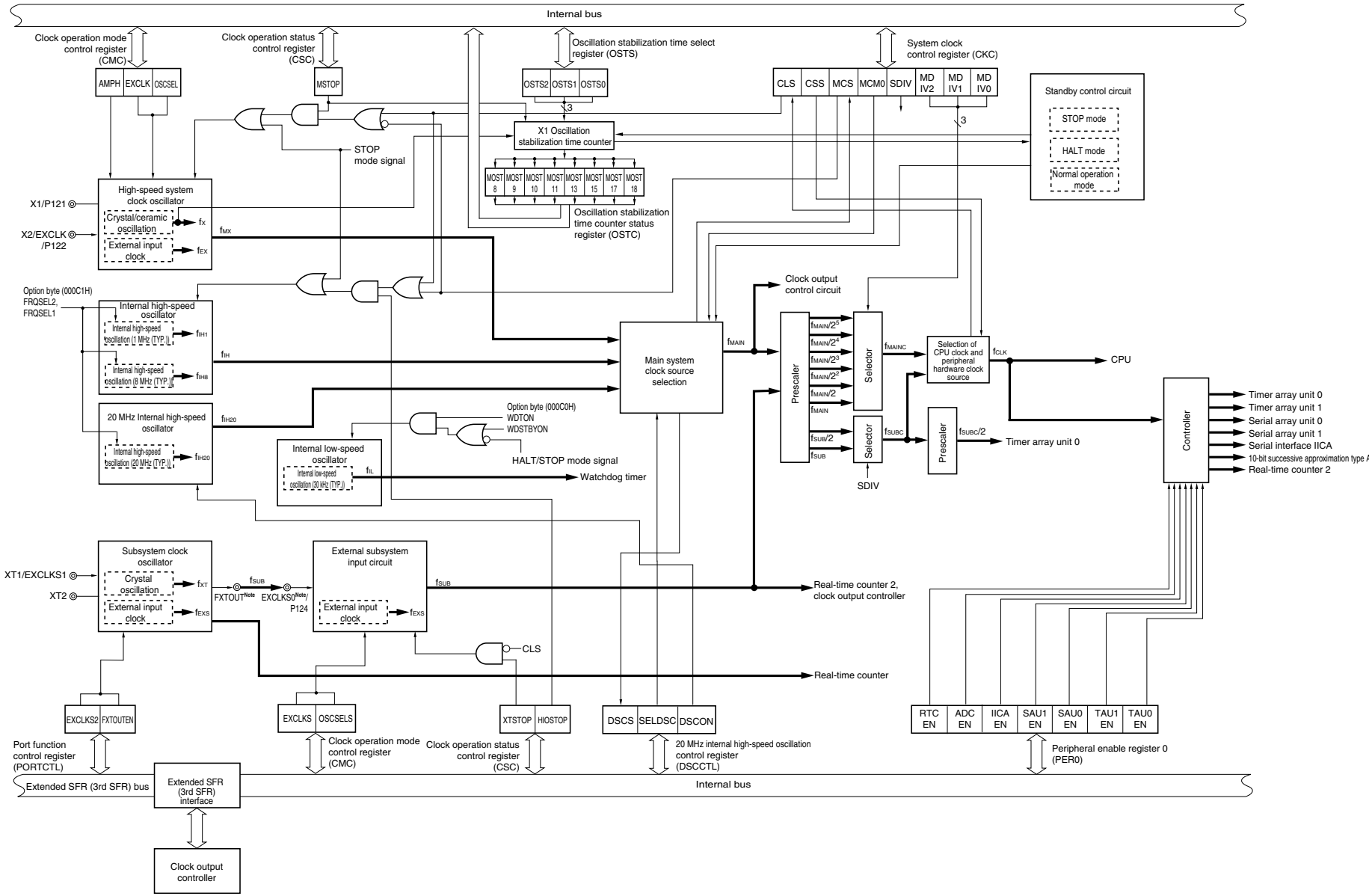
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) Port function control register (PORTCTL) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) System clock control register (CKC) 20 MHz internal high-speed oscillation control register (DSCCTL) Peripheral enable registers 0 (PER0) Operation speed mode control register (OSMC)
Oscillators	X1 oscillator XT1 oscillator Internal high-speed oscillator Internal low-speed oscillator

Figure 5-1. Block Diagram of Clock Generator



See **Figure 5-17 Block Diagram of Clock Output Controller**

Note For EXCLKS0 and FXTOUT pins, the output from FXTOUT pin must be connected to the input to P124/EXCLKS0 pin on the used board.

Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{IH1}: 1 MHz internal high-speed oscillation clock frequency
f_{IH8}: 8 MHz internal high-speed oscillation clock frequency
f_{IH20}: 20 MHz internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{MAINC}: Main system selection clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXS}: External subsystem clock oscillation frequency
f_{SUB}: Subsystem clock frequency
f_{SUBC}: Subsystem selection clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Internal low-speed oscillation clock frequency

5.3 Registers Controlling Clock Generator

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

- Clock operation mode control register (CMC)
- Port function control register (PORTCTL)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- System clock control register (CKC)
- 20 MHz internal high-speed oscillation control register (DSCCTL)
- Peripheral enable register 0 (PER0)
- Operation speed mode control register (OSMC)

(1) Clock operation mode control register (CMC)

This register is used to set the operation mode of the high-speed system clock and subsystem clock.

CMC can be written only once by an 8-bit memory manipulation instruction after reset release. This register can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

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

Address: FFFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS[Note]	OSCSELS[Note]	0	0	0	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

AMPH	Control of high-speed system clock oscillation frequency
0	2 MHz ≤ f_{MX} ≤ 10 MHz
1	10 MHz < f_{MX} ≤ 20 MHz

Note For the setting of EXCLKS and OSCSELS, see **(4) Setting of operation mode for subsystem clock pin**.

Remark f_{MX}: High-speed system clock frequency

Cautions 1. CMC can be written only once after reset release, by an 8-bit memory manipulation instruction.

2. After reset release, set CMC before X1 oscillation is started as set by the clock operation status control register (CSC) or XT1 oscillation is started as set by the port function control register (PORTCTL).

3. Be sure to set AMPH to 1 if the X1 clock oscillation frequency exceeds 10 MHz.

4. To use CMC with its initial value (00H), be sure to set it to 00H after releasing reset in order to prevent malfunction when a program loop occurs.

5. Be sure to clear bits 1 to 3 to "0".

(2) Port function control register (PORTCTL)

This register controls the operation mode of subsystem clock.

PORTCTL is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 5-3. Format of Port Function Control Register (PORTCTL)

Address: 91H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PORTCTL	SUBSTAT	0	0	0	0	0	EXCLKS2	FXTOUTEN

SUBSTAT	Oscillation stabilization status flag for subsystem clock
0	Pin reset status
1	After a pin reset release, this flag is set after the oscillation stabilization time for the subsystem clock (2^{16} (2 sec at 32.768 kHz)) elapses.

EXCLKS2	Set up subsystem clock
0	Connects a resonator.
1	Inputs an external clock signal.

FXTOUTEN	Specify whether to enable or disable output from subsystem clock oscillator
0	Disables output from subsystem clock oscillator.
1	Enables output from subsystem clock oscillator.

Cautions 1. Only rewrite the value of EXCLKS2 after clearing the FXTOUTEN and the RTCE bit of the RTCC0 register to 0 (FXTOUTEN = 0, RTCE = 0).

2. After a reset release, the resonator connection mode (EXCLKS2 = 0) cannot be specified after specifying the external clock input mode (EXCLKS2 = 1).

3. If the resonator connection mode is selected, be sure to set FXTOUTEN to 1 (to enable sub-clock oscillation) and RTCE to 1 (to enable RTC operation) after the oscillation stabilization time for the subsystem clock has elapsed (SUBSTAT = 1).

(3) Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock, internal high-speed oscillation clock, and subsystem clock (except the 20 MHz internal high-speed oscillation clock and internal low-speed oscillation clock). CSC can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to C0H.

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

Address: FFFA1H After reset: C0H R/W

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

MSTOP	High-speed system clock operation control		
	X1 oscillation mode	External clock input mode	Input port mode
0	X1 oscillator operating	External clock from EXCLK pin is valid	_
1	X1 oscillator stopped	External clock from EXCLK pin is invalid	

XTSTOP	Subsystem clock operation control
	External clock input mode
0	External clock from EXCLKS0 pin is valid
1	External clock from EXCLKS0 pin is invalid

HIOSTOP	Internal high-speed oscillation clock operation control
0	Internal high-speed oscillator operating
1	Internal high-speed oscillator stopped

Cautions 1. After reset release, set the clock operation mode control register (CMC) before starting X1 oscillation as set by MSTOP or enabling the external clock input as set by XTSTOP.

2. To start X1 oscillation as set by MSTOP, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).

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

4. The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) and the condition before clock oscillation is to be stopped are as follows.

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

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock External main system clock	CPU and peripheral hardware clocks operate with a clock other than the high-speed system clock. (• CLS = 0 and MCS = 0 • CLS = 1)	MSTOP = 1
Subsystem clock	CPU and peripheral hardware clocks operate with a clock other than the subsystem clock. (CLS = 0)	XTSTOP = 1
Internal high-speed oscillation clock	CPU and peripheral hardware clocks operate with a clock other than the internal high-speed oscillator clock and 20 MHz internal high-speed oscillation clock. (• CLS = 0 and MCS = 1 • CLS = 1)	HIOSTOP = 1

(4) Setting of operation mode for subsystem clock pin

The operation mode for the subsystem clock pin can be set by using bits 5 and 4 (EXCLKS and OSCSELS) of the clock operation mode control register (CMC) and bits 1 and 0 (EXCLKS2 and FXTOUTEN) of the port function control register (PORTCTL) in combination.

Table 5-3. Whether Subsystem Clock Can Be Used

CMC register		Subsystem Clock
Bit 5	Bit 4	
EXCLKS	OSCSELS	
0	0	Using disabled[Note]
0	1	Setting prohibited
1	0	
1	1	Using enabled

Note Only the real-time counter can be used.

Caution Confirm that bit 7 (CLS) of the system clock control register (CKC) is 0 (CPU is operating with main system clock) when changing the current values of EXCLKS and OSCSELS.

Table 5-4. Whether Subsystem Clock Can Be Used and Setting of Operation Mode for Subsystem Clock Pin

PORTCTL register		Subsystem Clock	Subsystem Clock Pin Operation Mode	XT1/EXCLKS1 Pin	XT2 Pin
Bit 1	Bit 0				
EXCLKS2	FXTOUTEN				
0	0	Using disabled[Note 1]	XT1 oscillation mode	Crystal resonator connection	
0	1	Using enabled			
1	0	Using disabled[Note 1]	External clock input mode	External clock input	**Note 2**
1	1	Using enabled			

Notes 1. Only the real-time counter can be used.

2. Leave the XT2 pin open when the XT1/EXCLKS1 pin is used as external system clock input.

Cautions 1. To use the subsystem clock, set EXCLKS and OSCSELS bits of CMC register and EXCLKS2 and FXTOUTEN bits of PORTCTL register.

2. Do not rewrite EXCLKS bit of CMC register and EXCLKS2 and FXTOUTEN bits of PORTCTL register when the CPU clock operates with the subsystem clock.

(5) Oscillation stabilization time counter status register (OSTC)

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

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

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

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

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

Remark The oscillation stabilization time counter starts counting in the following cases.

- When oscillation of the X1 clock starts (EXCLK, OSCSEL = 0, 1 → MSTOP = 0)
- When the STOP mode is released

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

Address: FFFA2H After reset: 00H R

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

MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	102.4 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	204.8 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	819.2 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	26.21 ms min.	13.11 ms min.

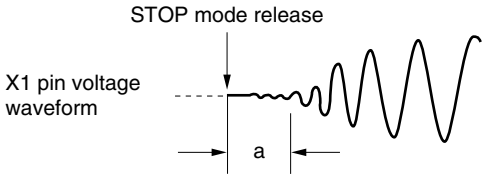
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS.

In the following cases, set the oscillation stabilization time of OSTS to the value greater than the count value which is to be checked by the OSTC register after the oscillation starts.

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

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



Remark fx: X1 clock oscillation frequency

(6) Oscillation stabilization time select register (OSTS)

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

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

When the internal high-speed oscillation clock is selected as the CPU clock, confirm with OSTC that the desired oscillation stabilization time has elapsed after the STOP mode is released. The oscillation stabilization time can be checked up to the time set using OSTC.

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

Reset signal generation sets OSTS to 07H.

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

Address: FFFA3H After reset: 07H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	Setting prohibited
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_X$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_X$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.21 ms	13.11 ms

Cautions 1. To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTS register before executing the STOP instruction.

2. Setting the oscillation stabilization time to 20 μs or less is prohibited.

3. To change the setting of the OSTS register, be sure to confirm that the counting operation of the OSTC register has been completed.

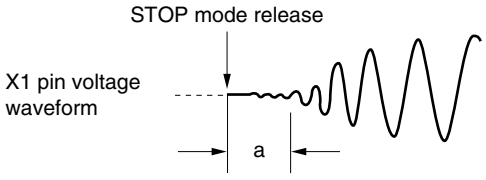
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

5. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS.

In the following cases, set the oscillation stabilization time of OSTS to the value greater than the count value which is to be checked by the OSTC register after the oscillation starts.

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

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



Remark fx: X1 clock oscillation frequency

(7) System clock control register (CKC)

This register is used to select a CPU/peripheral hardware clock and a division ratio.

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

Reset signal generation sets this register to 09H.

Figure 5-7. Format of System Clock Control Register (CKC)

Address: FFFA4H After reset: 09H R/W[Note 1]

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	SDIV	MDIV2	MDIV1	MDIV0

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

MCS	Status of Main system clock (f_{MAIN})
0	Internal high-speed oscillation clock (f_{IH}) or 20 MHz internal high-speed oscillation clock (f_{IH20})
1	High-speed system clock (f_{MX})

CSS	MCM0	SDIV	MDIV2	MDIV1	MDIV0	Selection of CPU/peripheral hardware clock (f_{CLK})
0	0	×	0	0	0	f_{IH}
		×	0	0	1	$f_{IH}/2$ (default)
		×	0	1	0	$f_{IH}/2^2$
		×	0	1	1	$f_{IH}/2^3$
		×	1	0	0	$f_{IH}/2^4$ [Note 2]
		×	1	0	1	$f_{IH}/2^5$ [Note 2]
0	1	×	0	0	0	f_{MX}
		×	0	0	1	$f_{MX}/2$
		×	0	1	0	$f_{MX}/2^2$
		×	0	1	1	$f_{MX}/2^3$
		×	1	0	0	$f_{MX}/2^4$
		×	1	0	1	$f_{MX}/2^5$ [Note 3]
1 [Note 4]	× [Note 4]	0	×	×	×	f_{SUB}
		1	×	×	×	$f_{SUB}/2$
Other than above						Setting prohibited

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

2. Setting is prohibited when f_{IH} = 1 MHz.

3. Setting is prohibited when f_{MX} < 4 MHz.

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

(**Remarks** and **Cautions** are listed on the next page.)

Remarks 1. f_{IH}: Internal high-speed oscillation clock frequency
f_{IH20}: 20 MHz Internal high-speed oscillation clock frequency
f_{MX}: High-speed system clock frequency
f_{SUB} Subsystem clock frequency
2. ×: don't care

Cautions 1. The clock set by CSS, MCM0, SDIV, and MDIV2 to MDIV0 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 counter 2, timer array unit (when f_{SUB}/2, f_{SUB}/4, the valid edge of TI0mn input, or the valid edge of INTRTCI is selected as the count clock), clock output, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral operating hardware clock.

2. If the peripheral hardware clock is used as the subsystem clock, the operations of 10-bit successive approximation type 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.

The fastest instruction can be executed in 1 clock of the CPU clock in the 78K0R/Lx3-M microcontrollers. Therefore, the relationship between the CPU clock (f_{CLK}) and the minimum instruction execution time is as shown in Table 5-5.

Table 5-5. Relationship Between CPU Clock and Minimum Instruction Execution Time

CPU Clock (Value set by the SDIV, and MDIV2 to MDIV0 bits)	Minimum Instruction Execution Time: $1/f_{CLK}$				
	Main System Clock (CSS = 0)				Subsystem Clock (CSS = 1)
	High-Speed System Clock (MCM0 = 1)		Internal High-Speed Oscillation Clock (MCM0 = 0)		
	At 10 MHz Operation	At 20 MHz Operation	At 8 MHz (TYP.) Operation	At 20 MHz (TYP.) Operation	At 32.768 kHz Operation
f_{MAIN}	0.1 μs	0.05 μs	0.125 μs (TYP.)	0.05 μs (TYP.)	–
$f_{MAIN}/2$	0.2 μs	0.1 μs	0.25 μs (TYP.) (default)	0.1 μs (TYP.)	–
$f_{MAIN}/2^2$	0.4 μs	0.2 μs	0.5 μs (TYP.)	0.2 μs (TYP.)	–
$f_{MAIN}/2^3$	0.8 μs	0.4 μs	1.0 μs (TYP.)	0.4 μs (TYP.)	–
$f_{MAIN}/2^4$	1.6 μs	0.8 μs	2.0 μs (TYP.)	0.8 μs (TYP.)	–
$f_{MAIN}/2^5$	3.2 μs	1.6 μs	4.0 μs (TYP.)	1.6 μs (TYP.)	–
f_{SUB}	–		–		30.5 μs
$f_{SUB}/2$	–		–		61 μs

Remark f_{MAIN}: Main system clock frequency (f_{IH}, f_{IH20}, or f_{MX})
f_{SUB}: Subsystem clock frequency

(8) 20 MHz internal high-speed oscillation control register (DSCCTL)

This register controls the 20 MHz internal high-speed oscillation clock (DSC) function.

It can be used to select whether to use the 20 MHz internal high-speed oscillation clock (f_{IH20}) as a peripheral hardware clock that supports 20 MHz.

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

Reset signal generation clears this register to 00H.

Figure 5-8. Format of 20 MHz Internal High-Speed Oscillation Control Register (DSCCTL)

Address: F00F6H After reset: 00H R/W[Note]

Symbol	7	6	5	4	<3>	<2>	1	<0>
DSCCTL	0	0	0	0	DSCS	SELDSC	0	DSCON

DSCS	20 MHz internal high-speed oscillation supply status flag
0	Not supplied
1	Supplied

SELDSC	Selection of 20 MHz internal high-speed oscillation for CPU/peripheral hardware clock (f_{CLK})
0	Does not select 20 MHz internal high-speed oscillation (clock selected by CKC register is supplied to f_{CLK})
1	Selects 20 MHz internal high-speed oscillation (20 MHz internal high-speed oscillation is supplied to f_{CLK})

DSCON	20 MHz internal high-speed oscillation clock (f_{IH20}) operation enable/disable
0	Disables operation.
1	Enables operation.

Note Bit 3 is read-only.

Cautions 1. 20 MHz internal oscillation can only be used if $V_{DD} \geq 2.7$ V.

2. Set SELDSC when 100 μs have elapsed after having set DSCON with $V_{DD} \geq 2.7$ V.

3. The internal high-speed oscillator must be operated (HIOSTOP = 0) when DSCON = 1.

(9) Peripheral enable register 0 (PER0)

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

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

Reset signal generation clears theses registers to 00H.

Figure 5-9. Format of Peripheral Enable Register 0 (PER0) (1/2)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of real-time counter 2 (RTC2) input clock[Note]
0	Stops input clock supply. • SFR used by the real-time counter 2 (RTC2) cannot be written. • The real-time counter 2 (RTC2) is in the reset status.
1	Supplies input clock. • SFR used by the real-time counter 2 (RTC2) can be read and written.

ADCEN	Control of 10-bit successive approximation type A/D converter input clock
0	Stops input clock supply. • SFR used by the 10-bit successive approximation type A/D converter cannot be written. • The 10-bit successive approximation type A/D converter is in the reset status.
1	Supplies input clock. • SFR used by the 10-bit successive approximation type A/D converter can be read and written.

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

SAU1EN	Control of serial array unit 1 input clock
0	Stops input clock supply. • SFR used by the serial array unit 1 cannot be written. • The serial array unit is in the reset status.
1	Supplies input clock. • SFR used by the serial array unit 1 can be read and written.

Note By using RTCEN, can supply and stop the clock that is used when accessing the real-time counter 2 (RTC2) from the CPU. RTCEN cannot control supply of the operating clock to RTC2.

Caution Be sure to set bit 6 to 0.

Figure 5-9. Format of Peripheral Enable Register 0 (PER0) (2/2)

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

TAU1EN	Control of timer array unit 1 input clock
0	Stops input clock supply. • SFR used by timer array unit 1 cannot be written. • Timer array unit 1 is in the reset status.
1	Supplies input clock. • SFR used by timer array unit 1 can be read and written.

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

(10) Operation speed mode control register (OSMC)

This register is used to control the step-up circuit of the flash memory for high-speed operation.

If the microcontroller operates at a low speed with a system clock of 10 MHz or less, the power consumption can be lowered by setting this register to the default value, 00H.

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

Reset signal generation clears this register to 00H.

Figure 5-10. Format of Operation Speed Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	0	0	0	FLPC	FSEL

RTCLPC	Setting in subsystem clock HALT mode
0	Enables subsystem clock supply to peripheral functions. (See Table 22-1 Operating Statuses in HALT Mode (2/3) for the peripheral functions whose operations are enabled.)
1	Stops subsystem clock supply to peripheral functions except real-time counter 2, clock output, and LCD controller/driver.

FLPC	FSEL	f_{CLK} frequency selection
0	0	Operates at a frequency of 10 MHz or less (default).
0	1	Operates at a frequency higher than 10 MHz.
1	0	Operates at a frequency of 1 MHz.
1	1	Setting prohibited

Cautions 1. Write "1" to FSEL before the following two operations.
- **Changing the clock prior to dividing f_{CLK} to a clock other than f_{IH}.**
- **Operating the DMA controller.**

2. The CPU waits (140.5 clock (f_{CLK})) when "1" is written to the FSEL bit.
Interrupt requests issued during a wait will be suspended.
However, counting the oscillation stabilization time of f_X can continue even while the CPU is waiting.

3. To increase f_{CLK} to 10 MHz or higher, set FSEL to "1", then change f_{CLK} after two or more clocks have elapsed.

4. Confirm that the clock is operating at 10 MHz or less before setting FSEL = 0.

5. To shift to STOP mode while $V_{DD} \leq 2.7$ V, set FSEL = 0 after setting f_{CLK} to 10 MHz or less.

6. The HALT mode current when operating on the subsystem clock can be reduced by setting RTCLPC to 1. However, the clock cannot be supplied to peripheral functions except the real-time counter 2 in the subsystem clock HALT mode. Set bit 7 (RTCEN) of PER0 to 1 and bits 0 to 6 of PER0 to 0 before setting the subsystem clock HALT mode.

7. Once FLPC has been set from 0 to 1, setting it back to 0 from 1 other than by reset is prohibited.

8. When setting FSEL to "1", do so while RMC = 00H.
When setting FLPC to "1", do so while RMC = 5AH.

5.4 System Clock Oscillator

5.4.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (2 to 20 MHz) connected to the X1 and X2 pins.

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

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

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

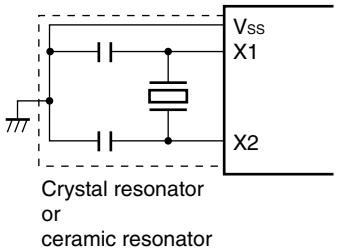
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

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

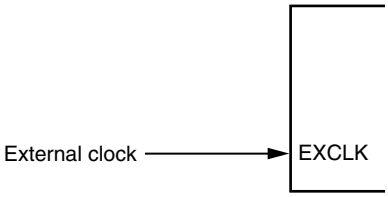
Figure 5-11 shows an example of the external circuit of the X1 oscillator.

Figure 5-11. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



Cautions are listed on the next page.

5.4.2 XT1 oscillator

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

Also, external clock can be input. In such case, input the clock signal to EXCLKS1 pin.

To use the XT1 oscillator, set bits 5 and 4 (EXCLKS, OSCSELS) of the clock operation mode control register (CMC) and bits 1 and 0 (EXCLKS2, FXTOUTEN) of the port function control register (PORTCTL) as follows.

- Crystal or ceramic oscillation: EXCLKS, OSCSELS = 1, 1, EXCLKS2, FXTOUTEN = 0, 1
- External clock input: EXCLKS, OSCSELS = 1, 1, EXCLKS2, FXTOUTEN = 1, 1

When the XT1 oscillator is not used, set the input port mode as follows.

- EXCLKS, OSCSELS = 0, 0, EXCLKS2, FXTOUTEN = 1, 0

Figure 5-12 shows an example of the external circuit of the XT1 oscillator.

Figure 5-12. Example of External Circuit of XT1 Oscillator (Crystal Oscillation)

Caution 1. When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-11 and 5-12 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 Vss. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

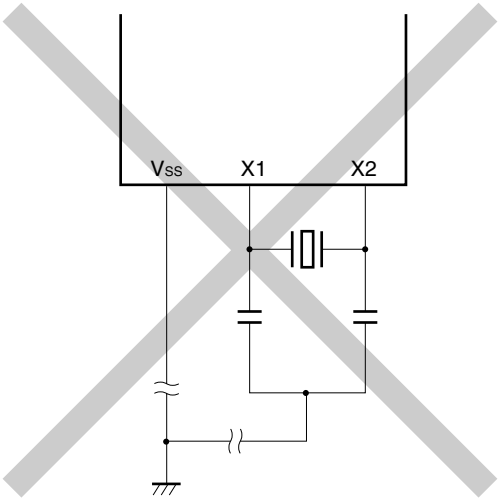
Note that the XT1 oscillator is designed as a low-gain circuit for achieving low-power consumption. Note the following points when designing the XT1 oscillator.

- **The pins and circuit board include parasitic capacitance. Therefore, confirm that there are no problems by performing oscillation evaluation on the circuit board to be actually used.**
- **When low-consumption oscillation or super-low-consumption oscillation is selected, lower power consumption than when selecting normal oscillation can be achieved. However, in this case, the XT1 oscillation margin is reduced, so perform sufficient oscillation evaluation of the resonator to be used for XT1 oscillation before using the resonator.**
- **Keep the wiring length between the XT1 and XT2 pins and resonator as short as possible and parasitic capacitance and wire resistance as small as possible.**
- **Configure the circuit board by using material with little parasitic capacitance and wire resistance.**
- **Place a ground pattern that has the same potential as Vss (if possible) around the XT1 oscillator.**
- **Do not cross the signal lines between the XT1 and XT2 pins and the resonator with other signal lines. Do not route the signal lines near a signal line through which a high fluctuating current flows.**
- **Moisture absorption by the circuit board and condensation on the board in a highly humid environment may cause the impedance between the XT1 and XT2 pins to drop and disable oscillation. When using the circuit board in such an environment, prevent the circuit board from absorbing moisture by taking measures such as coating the circuit board.**
- **Coat the surface of the circuit board by using material that does not generate capacitance or leakage between the XT1 and XT2 pins.**

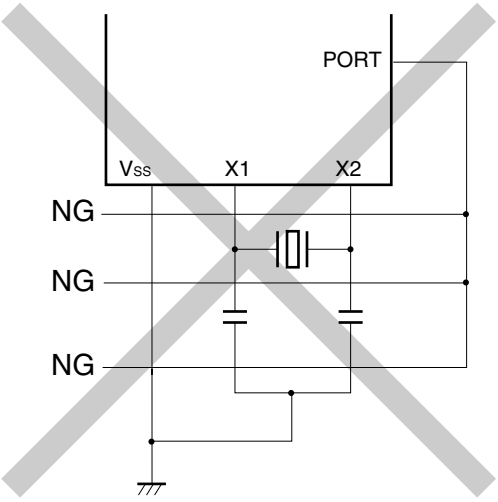
Figure 5-13 shows examples of incorrect resonator connection.

Figure 5-13. Examples of Incorrect Resonator Connection (1/2)

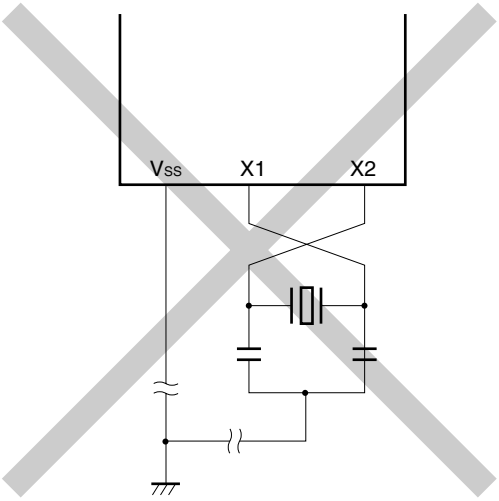
(a) Too long wiring



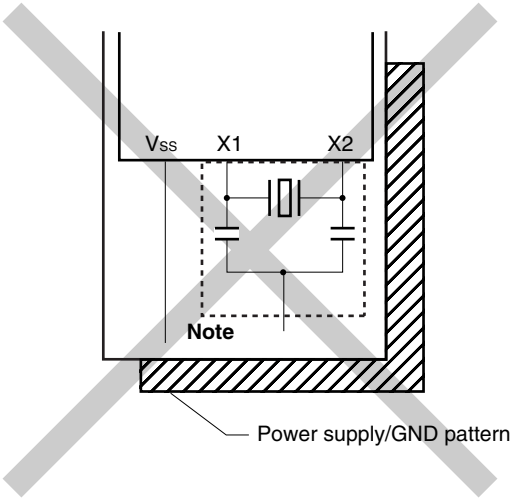
(b) Crossed signal line



(c) Signal lines of X1 and X2 cross



(d) Power supply/GND pattern exists underneath X1 and X2 wiring

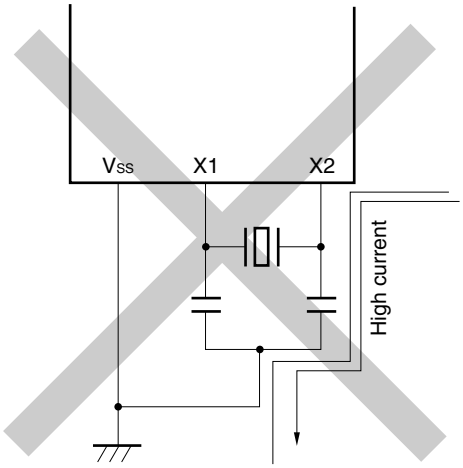


Note Do not place a power supply/GND pattern underneath the wiring section (in broken lines above) of the X1 and X2 pins and resonator in the multilayer board and double-sided board.
Do not configure a layout that may cause capacitance elements and affect the oscillation characteristics.

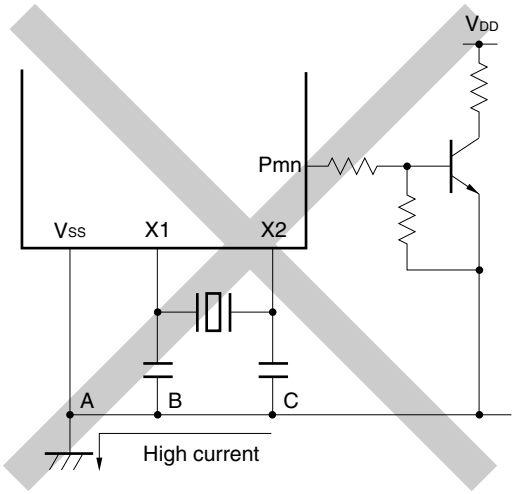
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 5-13. Examples of Incorrect Resonator Connection (2/2)

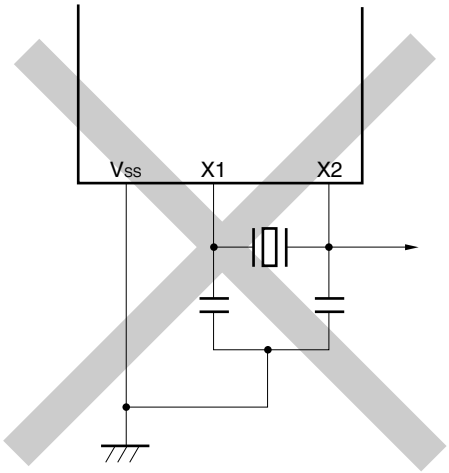
(e) Wiring near high alternating current



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



(g) Signals are fetched



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

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

5.4.3 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0R/Lx3-M microcontroller (1, 8 and 20 MHz (TYP.)). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC) and bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL).

Caution To use the 1, 8, or 20 MHz internal high-speed oscillation clock, use the option byte to set the frequency in advance (for details, see CHAPTER 27 OPTION BYTE). Also, the internal high-speed oscillator automatically starts oscillating after reset release. (If 8 MHz or 20 MHz is selected by using the option byte, the microcontroller operates using the 8 MHz internal high-speed oscillator.) To use the 20 MHz internal high-speed oscillator to operate the microcontroller, oscillation is started by setting bit 0 (DSCON) of the DSCCTL register to 1 with $V_{DD} \geq 2.7$ V.

5.4.4 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0R/Lx3-M microcontroller.

The internal low-speed oscillation clock is used only as the watchdog timer clock. The internal low-speed oscillation clock cannot be used as the CPU clock.

After a reset release, the internal low-speed oscillator automatically starts oscillation, and the watchdog timer is driven (30 kHz (TYP.)) if the watchdog timer operation is enabled by the option byte.

The internal low-speed oscillator continues oscillation except when the watchdog timer stops. When the watchdog timer operates, the internal low-speed oscillation clock does not stop, even in case of a program loop.

5.4.5 Prescaler

The prescaler generates a CPU/peripheral hardware clock by dividing the main system clock and subsystem clock.

5.5 Clock Generator Operation

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

- Main system clock f_{MAIN}
  - High-speed system clock f_{MX}
    X1 clock f_X
    External main system clock f_{EX}
  - Internal high-speed oscillation clock f_{IH}
    1 MHz internal high-speed oscillation clock f_{IH1}
    8 MHz internal high-speed oscillation clock f_{IH8}
    20 MHz internal high-speed oscillation clock f_{IH20}
- Subsystem clock f_{SUB}
  - XT1 clock f_{XT}
  - External subsystem clock f_{EXS}
- Subsystem selection clock f_{SUBC}
- Internal low-speed oscillation clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}

The CPU starts operation when the internal high-speed oscillator starts outputting after a reset release in the 78K0R/Lx3-M microcontrollers, thus enabling the following.

(1) Enhancement of security function

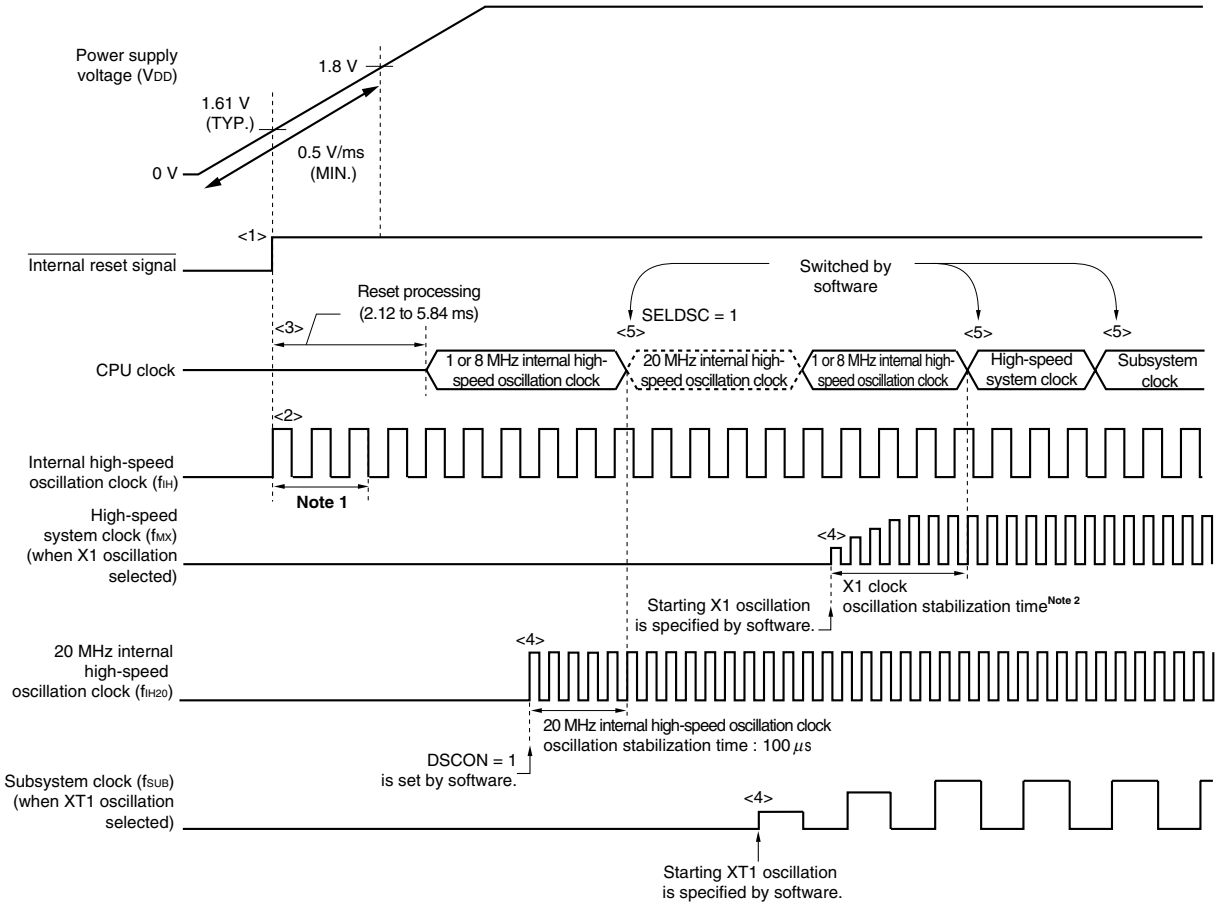
When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the internal high-speed oscillation clock, so the device can be started by the internal high-speed oscillation clock after a reset release. As a result, reset sources can be detected by software and the minimum amount of safety processing can be done during anomalies to ensure that the system terminates safely.

(2) Improvement of performance

Because the CPU can be started without waiting for the X1 clock oscillation stabilization time, the total performance can be improved.

When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-14 and Figure 5-15.

Figure 5-14. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1))



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

<2> When the power supply voltage exceeds 1.61 V (TYP.), the reset is released and the internal high-speed oscillator automatically starts oscillation.

<3> The CPU starts operation on the internal high-speed oscillation clock Note 3 after a reset processing such as waiting for the voltage of the power supply or regulator to stabilize has been performed after reset release.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **(1)** in **5.6.1 Example of controlling high-speed system clock** and **(1)** in **5.6.3 Example of controlling subsystem 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 **(3)** in **5.6.1 Example of controlling high-speed system clock** and **(3)** in **5.6.3 Example of controlling subsystem clock**).

Switch to the 20 MHz internal high-speed oscillation clock by setting the DSCON bit (bit 0 of the 20 MHz internal high-speed oscillation control register (DSCCTL)), waiting for 100 μs, and then setting the SELDSC bit to 1 by using software Note 4.

(**Notes** and **Cautions** are listed on the next page.)

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

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

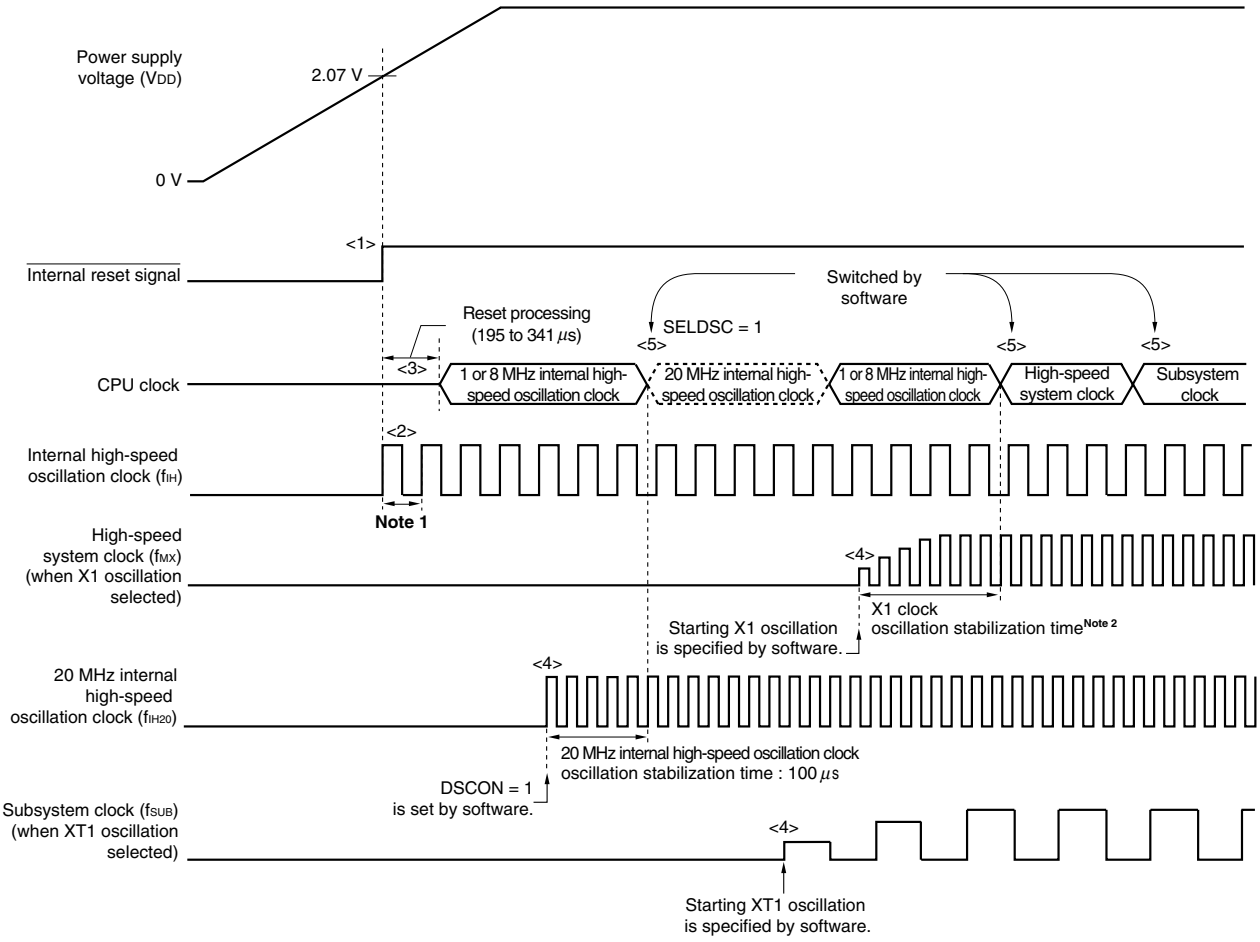
3. The microcontroller operates on the 8 MHz internal high-speed oscillation clock if 8 MHz or 20 MHz is selected for the internal high-speed oscillator by using the option byte or on the 1 MHz internal high-speed oscillation clock if 1 MHz is selected.

4. If the internal high-speed oscillator is set to 1 MHz by using the option byte, the 20 MHz internal high-speed oscillation clock cannot be used.

Cautions 1. If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 1.8 V, input a low level to the RESET pin from power application until the voltage reaches 1.8 V, or set the LVI default start function stopped by using the option byte (LVIOFF = 0) (see Figure 5-15). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 5-14 after reset release by the RESET pin.

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

Figure 5-15. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0))



<1> When the power is turned on, an internal reset signal is generated by the low-voltage detector (LVI) circuit.

<2> When the power supply voltage exceeds 2.07 V (TYP.), the reset is released and the internal high-speed oscillator Note 3 automatically starts oscillation.

<3> After the reset is released and reset processing is performed, the CPU starts operation on the internal high-speed oscillation clock Note 3.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **(1)** in **5.6.1 Example of controlling high-speed system clock** and **(1)** in **5.6.3 Example of controlling subsystem clock**).

Switch to oscillation using the 20 MHz internal high-speed oscillation clock after setting the DSCON bit to 1 by using software.

<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 **(3)** in **5.6.1 Example of controlling high-speed system clock** and **(3)** in **5.6.3 Example of controlling subsystem clock**).

Switch to the 20 MHz internal high-speed oscillation clock after confirming that the power supply voltage is at least 2.7 V, setting the DSCON bit (bit 0 of the 20 MHz internal high-speed oscillation control register (DSCCTL)), waiting for 100 μs, and then setting the SELDSC bit to 1 by using software Note 4.

(**Notes** and **Cautions** are listed on the next page.)

Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation 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. The microcontroller operates on the 8 MHz internal high-speed oscillation clock if 8 MHz or 20 MHz is selected for the internal high-speed oscillator by using the option byte or on the 1 MHz internal high-speed oscillation clock if 1 MHz is selected.

4. If the internal high-speed oscillator is set to 1 MHz by using the option byte, the 20 MHz internal high-speed oscillation clock cannot be used.

Cautions 1. A voltage stabilization time (about 2.12 to 5.84 ms) is required after the supply voltage reaches 1.61 V (TYP.). If the time for the supply voltage to rise from 1.61 V (TYP.) to 2.07 V (TYP.) is shorter than the voltage stabilization time, reset processing is entered after the voltage stabilization time elapses.

2. 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 controlling high-speed system clock

The following two types of high-speed system clocks are available.

- X1 clock: Crystal/ceramic resonator is connected to the X1 and X2 pins.
- External main system clock: External clock is input to the EXCLK pin.

When the high-speed system clock is not used, the X1/P121 and X2/EXCLK/P122 pins can be used as input port pins.

Caution The X1/P121 and X2/EXCLK/P122 pins are in the input port mode after a reset release.

The following describes examples of setting procedures for the following cases.

(1) When oscillating X1 clock
(2) When using external main system clock
(3) When using high-speed system clock as CPU/peripheral hardware clock
(4) When stopping high-speed system clock

(1) Example of setting procedure when oscillating the X1 clock

<1> Setting P121/X1 and P122/X2/EXCLK pins and setting oscillation frequency (CMC register)

- 2 MHz ≤ fx ≤ 10 MHz

EXCLK	OSCSEL	EXCLKS	OSCSELS	0	0	0	AMPH
0	1	0	0	0	0	0	0

- 10 MHz < fx ≤ 20 MHz

EXCLK	OSCSEL	EXCLKS	OSCSELS	0	0	0	AMPH
0	1	0	0	0	0	0	1

Remarks 1. fx: X1 clock oscillation frequency

2. For setting of the XT1 and XT2 pins, see **5.6.3 Example of controlling subsystem clock**.

<2> Controlling oscillation of X1 clock (CSC register)
If MSTOP is cleared to 0, the X1 oscillator starts oscillating.

<3> Waiting for the stabilization of the oscillation of X1 clock
Check the OSTC register and wait for the necessary time.
During the wait time, other software processing can be executed with the internal high-speed oscillation clock.

Cautions 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction.
Therefore, it is necessary to also set the value of the OSCSELS bit at the same time. For OSCSELS bit, see 5.6.3 Example of controlling subsystem clock.

2. Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 32 ELECTRICAL SPECIFICATIONS).

(2) Example of setting procedure when using the external main system clock

<1> Setting P121/X1 and P122/X2/EXCLK pins (CMC register)

EXCLK	OSCSEL	EXCLKS	OSCSELS	0	0	0	AMPH
1	1	0	0	0	0	0	0/1

Remark For setting of the XT1 and XT2 pins, see **5.6.3 (1) Example of setting procedure when oscillating XT1 clock**.

<2> Controlling external main system clock input (CSC register)
When MSTOP is cleared to 0, the input of the external main system clock is enabled.

Cautions 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction.
Therefore, it is necessary to also set the value of the OSCSELS bits at the same time. For OSCSELS bits, see 5.6.3 Example of controlling subsystem clock.
2. Set the external main system clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 32 ELECTRICAL SPECIFICATIONS).

(3) Example of setting procedure when using high-speed system clock as CPU/peripheral hardware clock

<1> Setting high-speed system clock oscillation[Note]
(See **5.6.1 (1) Example of setting procedure when oscillating the X1 clock** and **(2) Example of setting procedure when using the external main system clock.**)

Note The setting of <1> is not necessary when high-speed system clock is already operating.

<2> Setting the high-speed system clock as the source clock of the CPU/peripheral hardware clock and setting the division ratio of the set clock (CKC register)

MCM0	MDIV2	MDIV1	MDIV0	Selection of CPU/Peripheral Hardware Clock (f_{CLK})
1	0	0	0	f_{MX}
	0	0	1	$f_{MX}/2$
	0	1	0	$f_{MX}/2^2$
	0	1	1	$f_{MX}/2^3$
	1	0	0	$f_{MX}/2^4$
	1	0	1	$f_{MX}/2^5$ [Note]

Note Setting is prohibited when f_{MX} < 4 MHz.

<3> If some peripheral hardware macros are not used, supply of the input clock to each hardware macro can be stopped.

(PER0 register)

RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

xxxEN	Input clock control
0	Stops input clock supply.
1	Supplies input clock.

Remark RTCEN: Control of the real-time counter 2 input clock
ADCEN: Control of the 10-bit successive approximation type A/D converter input clock
IICAEN: Control of the serial interface IICA input clock
SAU1EN: Control of the serial array unit 1 unit input clock
SAU0EN: Control of the serial array unit 0 unit input clock
TAU1EN: Control of the timer array unit 1 input clock
TAU0EN: Control of the timer array unit 0 input clock

(4) Example of setting procedure when stopping the high-speed system clock

The high-speed system clock can be stopped (disabling clock input if the external clock is used) in the following two ways.

- Executing the STOP instruction
- Setting MSTOP to 1

(a) To execute a STOP instruction

<1> Setting to stop peripheral hardware

Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 22 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after STOP mode is released

If the X1 clock oscillates before the STOP mode is entered, set the value of the OSTS register before executing the STOP instruction.

<3> Executing the STOP instruction

When the STOP instruction is executed, the system is placed in the STOP mode and X1 oscillation is stopped (the input of the external clock is disabled).

(b) To stop X1 oscillation (disabling external clock input) by setting MSTOP to 1

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the high-speed system clock. When CLS = 0 and MCS = 1, the high-speed system clock is supplied to the CPU, so change the CPU clock to the subsystem clock or internal high-speed oscillation clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock or 20 MHz internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Setting of X1 clock oscillation stabilization time after restart of X1 clock oscillation[Note]

Prior to setting "1" to MSTOP, set the OSTS register to a value greater than the count value to be confirmed with the OSTS register after X1 clock oscillation is restarted.

<3> Stopping the high-speed system clock (CSC register)

When MSTOP is set to 1, X1 oscillation is stopped (the input of the external clock is disabled).

Note This setting is required to resume the X1 clock oscillation when the high-speed system clock is in the X1 oscillation mode.

This setting is not required in the external clock input mode.

Caution Be sure to confirm that MCS = 0 or CLS = 1 when setting MSTOP to 1. In addition, stop peripheral hardware that is operating on the high-speed system clock.

5.6.2 Example of controlling internal high-speed oscillation clock

The following describes examples of clock setting procedures for the following cases.

(1) When restarting oscillation of the internal high-speed oscillation clock
(2) When using internal high-speed oscillation clock as CPU/peripheral hardware clock
(3) When stopping the internal high-speed oscillation clock

(1) Example of setting procedure when restarting oscillation of the internal high-speed oscillation clock[Note]

<1> Setting restart of oscillation of the internal high-speed oscillation clock (CSC register)

When HIOSTOP is cleared to 0, the internal high-speed oscillation clock restarts oscillation.

Note After a reset release, the internal high-speed oscillator automatically starts oscillating and the internal high-speed oscillation clock is selected as the CPU/peripheral hardware clock.

(2) Example of setting procedure when using internal high-speed oscillation clock as CPU/peripheral hardware clock

<1> Restarting oscillation of the internal high-speed oscillation clock[Note]

(See **5.6.2 (1) Example of setting procedure when restarting internal high-speed oscillation clock**).

Note The setting of <1> is not necessary when the internal high-speed oscillation clock is operating.

<2> Setting the internal high-speed oscillation clock as the source clock of the CPU/peripheral hardware clock and setting the division ratio of the set clock (CKC register)

MCM0	MDIV2	MDIV1	MDIV0	Selection of CPU/Peripheral Hardware Clock (f_{CLK})
0	0	0	0	f_{IH}
	0	0	1	$f_{IH}/2$
	0	1	0	$f_{IH}/2^2$
	0	1	1	$f_{IH}/2^3$
	1	0	0	$f_{IH}/2^4$ Note
	1	0	1	$f_{IH}/2^5$ Note

Note Setting is prohibited when f_{IH} = 1 MHz.

Caution If switching the CPU/peripheral hardware clock from the high-speed system clock to the internal high-speed oscillation clock after restarting the internal high-speed oscillation clock, do so after 10 μs or more have elapsed.
If the switching is made immediately after the internal high-speed oscillation clock is restarted, the accuracy of the internal high-speed oscillation cannot be guaranteed for 10 μs.

(3) Example of setting procedure when stopping the internal high-speed oscillation clock

The internal high-speed oscillation clock can be stopped in the following two ways.

- Executing the STOP instruction
- Setting HIOSTOP to 1

(a) To execute a STOP instruction

<1> Setting of peripheral hardware

Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 22 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after STOP mode is released

If the X1 clock oscillates before the STOP mode is entered, set the value of the OSTS register before executing the STOP instruction.

<3> Executing the STOP instruction

When the STOP instruction is executed, the system is placed in the STOP mode and internal high-speed oscillation clock is stopped.

(b) To stop internal high-speed oscillation clock by setting HIOSTOP to 1

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the internal high-speed oscillation clock.

When CLS = 0 and MCS = 0, the internal high-speed oscillation clock is supplied to the CPU, so change the CPU clock to the high-speed system clock or subsystem clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock or 20 MHz internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the internal high-speed oscillation clock (CSC register)
When HIOSTOP is set to 1, internal high-speed oscillation clock is stopped.

Caution Be sure to confirm that MCS = 1 or CLS = 1 when setting HIOSTOP to 1. In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock.

5.6.3 Example of controlling subsystem clock

The subsystem clock has following crystal resonator connections.

- XT1 clock: Crystal/ceramic resonator is connected across the XT1 and XT2 pins.
- External subsystem clock: External clock is input to the EXCLKS1 pin.

Caution Do not start the peripheral hardware operation with the external clock from peripheral hardware pins when the internal high-speed oscillation clock and high-speed system clock are stopped while the CPU operates with the subsystem clock, or when in the STOP mode.

The following describes examples of setting procedures for the following cases.
(1) When oscillating XT1 clock
(2) When using external subsystem clock
(3) When using subsystem clock as CPU clock
(4) When stopping subsystem clock

Caution When the subsystem clock is used as the CPU clock, the subsystem clock is also supplied to the peripheral hardware (except the real-time counter 2, timer array unit (when f_{SUB}/2, f_{SUB}/4, the valid edge of TI0mn input, or the valid edge of INTRTCI is selected as the count clock), clock output, and watchdog timer). At this time, the operations of the 10-bit successive approximation type 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.

(1) Example of setting procedure when oscillating the XT1 clock

<1> Setting XT1 and XT2 pins and selecting operation mode (PORTCTL register)
When EXCLKS2 and FXTOUTEN are set as any of the following, the mode is switched from port mode to XT1 oscillation mode.

EXCLKS2	FXTOUTEN	Operation Mode of Subsystem Clock Pin	XT1/EXCLKS1 Pins	XT2 Pin
0	0	XT1 oscillation mode	Crystal/ceramic resonator connection	
0	1			

Remark For setting of the P121/X1 and P122/X2 pins, see **5.6.1 Example of controlling high-speed system clock**.

<2> Waiting for the stabilization of the subsystem clock oscillation
Wait for the oscillation stabilization time of the subsystem clock by software, using a timer function.

<3> Setting the operation mode (PORTCTL register)
Set 1 to FXTOUTEN bit of PORTCTL register.

<4> Setting the operation mode (CMC register)
Set 1 to EXCLKS and OSCSELS bit of CMC register.

<5> Setting the external clock (CSC register)
Set 0 to XTSTOP bit of CSC register.

Cautions 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction.
Therefore, it is necessary to also set the value of the EXCLK and OSCSEL bits at the same time. For EXCLK and OSCSEL bits, see 5.6.1 (1) Example of setting procedure when oscillating the X1 clock or 5.6.1 (2) Example of setting procedure when using the external main system clock.
2. Do not change the values of EXCLKS2 and FXOUTEN while the subsystem clock is operating.

(2) Example of setting procedure when using the external subsystem clock

<1> Setting XT1 and XT2 pins, selecting XT1 clock/external clock and controlling oscillation (PORTCTL registers)
When EXCLKS2 and FXTOUTEN are set as any of the following, the mode is switched from port mode to external clock input mode. In this case, input the external clock to the EXCLKS1/XT1 pins.

EXCLKS2	FXTOUTEN	Operation Mode of Subsystem Clock Pin	XT1/EXCLKS1 Pins	XT2 Pin
1	0	External clock input mode	External clock input mode	**Note**
1	1			

Note Leave the XT2 pin open when the XT1/EXCLKS1 pin is used as external system clock input.

Caution Do not change the values of EXCLKS2 and FXOUTEN while the subsystem clock is operating.

<2> Setting the operation mode (PORTCTL register)
Set 1 to FXTOUTEN bit of PORTCTL register.

<3> Setting the operation mode (CMC register)
Set 1 to EXCLKS and OSCSELS bit of CMC register.

<4> Setting the external clock (CSC register)
Set 0 to XTSTOP bit of CSC register.

(3) Example of setting procedure when using the subsystem clock as the CPU clock

<1> Setting subsystem clock oscillation[Note]

(See **5.6.3 (1) Example of setting procedure when oscillating the XT1 clock** and **(2) Example of setting procedure when using the external subsystem clock**.)

Note The setting of <1> is not necessary when while the subsystem clock is operating.

<2> Setting the subsystem clock as the source clock of the CPU/peripheral hardware clock and setting the division ratio of the set clock (CKC register)

CSS	SDIV	Selection of CPU/Peripheral Hardware Clock (f_{CLK})
1	0	f_{SUB}
	1	$f_{SUB}/2$

Caution When the subsystem clock is used as the CPU clock, the subsystem clock is also supplied to the peripheral hardware (except the real-time counter 2, timer array unit (when $f_{SUB}/2$, $f_{SUB}/4$, the valid edge of TI0mn input, or the valid edge of INTRTCI is selected as the count clock), clock output, and watchdog timer). At this time, the operations of the 10-bit successive approximation type 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.

(4) Example of setting procedure when stopping the subsystem clock

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the subsystem clock.

When CLS = 1, the subsystem clock is supplied to the CPU, so change the CPU clock to the internal high-speed oscillation clock or high-speed system clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock or 20 MHz internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the subsystem clock (CSC register)

When XTSTOP is set to 1, subsystem clock is stopped.

Cautions 1. Be sure to confirm that CLS = 0 when setting XTSTOP to 1. In addition, stop the peripheral hardware if it is operating on the subsystem clock.

2. The subsystem clock oscillation cannot be stopped using the STOP instruction.

5.6.4 Example of controlling internal low-speed oscillation clock

The internal low-speed oscillation clock cannot be used as the CPU clock. Used only as the watchdog timer clock.

The internal low-speed oscillator automatically starts oscillation after a reset release, and the watchdog timer is driven (30 kHz (TYP.)) if the watchdog timer operation is enabled by the option byte.

The internal low-speed oscillator continues oscillation except when the watchdog timer stops. When the watchdog timer operates, the internal low-speed oscillation clock does not stop even in case of a program loop.

(1) Example of setting procedure when stopping the internal low-speed oscillation clock

The internal low-speed oscillation clock can be stopped in the following two ways.

- Stop the watchdog timer in the HALT/STOP mode by the option byte (bit 0 (WDSTBYON) of 000C0H = 0), and execute the HALT or STOP instruction.
- Stop the watchdog timer by the option byte (bit 4 (WDTON) of 000C0H = 0).

(2) Example of setting procedure when restarting oscillation of the internal low-speed oscillation clock

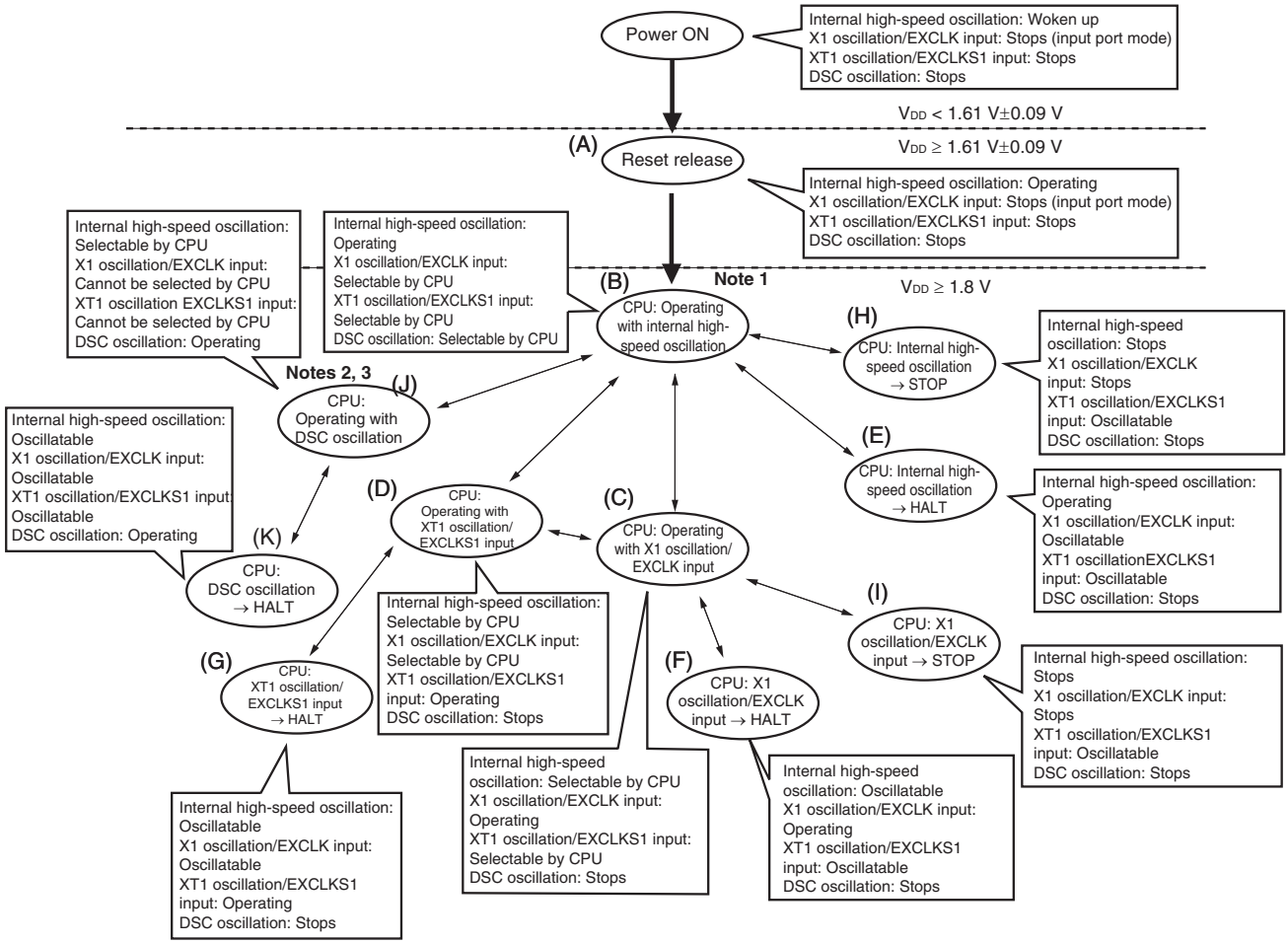
The internal low-speed oscillation clock can be restarted as follows.

- Release the HALT or STOP mode
  (only when the watchdog timer is stopped in the HALT/STOP mode by the option byte (bit 0 (WDSTBYON) of 000C0H) = 0) and when the watchdog timer is stopped as a result of execution of the HALT or STOP instruction).

5.6.5 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 Diagram



Notes 1. After reset release, an operation at one of the following operating frequencies is started, because $f_{CLK} = f_{IH}/2$ has been selected by setting the system clock control register (CKC) to 09H.

- When 1 MHz has been selected by using the option byte: 500 kHz (1 MHz/2)
- When 8 MHz or 20 MHz has been selected by using the option byte: 4 MHz (8 MHz/2)

2. Specify 20 MHz internal oscillation after checking that V_{DD} is at least 2.7 V.

3. 20 MHz internal oscillation cannot be used if 1 MHz internal oscillation is selected by using the option byte.

Remarks 1. If the low-power-supply detector (LVI) is set to ON by default by the option bytes, the reset will not be released until the power supply voltage (V_{DD}) exceeds 2.07 V ±0.2 V.
After the reset operation, the status will shift to (B) in the above figure.

2. DSC: 20 MHz internal high-speed oscillation clock

Table 5-6 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-6. CPU Clock Transition and SFR Register Setting Examples (1/6)

(1) CPU operating with internal high-speed oscillation clock (B) after reset release (A)

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

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

(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

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

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

2. FSEL = 1 when f_{CLK} > 10 MHz
If a divided clock is selected and f_{CLK} ≤ 10 MHz, use with FSEL = 0 is possible even if fx > 10 MHz.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 32 ELECTRICAL SPECIFICATIONS).

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

(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	PORTCTL Register		CMC Register[Note]		CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS2	FXTOUTEN	EXCLKS	OSCSELS	XTSTOP		CSS
(A) → (B) → (D) (XT1 clock)	0	1	1	1	0	Necessary	1
(A) → (B) → (D) (External subsystem clock)	1	1	1	1	0	Unnecessary	1

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

Remark (A) to (K) in Table 5-6 correspond to (A) to (K) in Figure 5-16.

Table 5-6. CPU Clock Transition and SFR Register Setting Examples (2/6)

(4) CPU operating with 20 MHz internal high-speed oscillation clock (J) after reset release (A)

(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	DSCCTL Register[Note]	Waiting for Oscillation Stabilization	DSCCTL Register
	DSCON		SELDSC
(A) → (B) → (J)	1	Necessary (100 μs)	1

Note Check that $V_{DD} \geq 2.7$ V and set DSCON = 1.

(5) CPU clock changing from internal high-speed oscillation clock (B) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note 1]			OSTS Register	CSC Register	OSMC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP	FSEL		MCM0
(B) → (C) (X1 clock: 2 MHz ≤ fX ≤ 10 MHz)	0	1	0	**Note 2**	0	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < fX ≤ 20 MHz)	0	1	1	**Note 2**	0	1[Note 3]	Must be checked	1
(B) → (C) (external main clock)	1	1	×	**Note 2**	0	0/1	Must not be checked	1

Unnecessary if these registers are already set

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

Notes 1. The CMC register can be changed only once after reset release. This setting is not necessary if it has already been set.

2. Set the oscillation stabilization time as follows.

- Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS

3. FSEL = 1 when f_{CLK} > 10 MHz

If a divided clock is selected and $f_{CLK} \leq 10$ MHz, use with FSEL = 0 is possible even if f_X > 10 MHz.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 32 ELECTRICAL SPECIFICATIONS).

Remarks 1. ×: don't care

2. (A) to (K) in Table 5-6 correspond to (A) to (K) in Figure 5-16.

Table 5-6. CPU Clock Transition and SFR Register Setting Examples (3/6)

(6) CPU clock changing from internal high-speed oscillation clock (B) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	PORTCTL Register		CMC Register[Note]		CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS2	FXTOUTEN	EXCLKS	OSCSELS	XTSTOP		CSS
(B) → (D) (XT1 clock)	0	1	1	1	0	Necessary	1
(B) → (D) (External subsystem clock)	1	1	1	1	0	Unnecessary	1

Unnecessary if the CPU is operating with the subsystem clock

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

(7) CPU clock changing from internal high-speed oscillation clock (B) to 20 MHz internal high-speed oscillation clock (J)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	DSCCTL Register[Note]	Waiting for Oscillation Stabilization	DSCCTL Register
	DSCON		SELDSC
(B) → (J)	1	Necessary (100 μs)	1

Unnecessary if the CPU is operating with the 20 MHz internal high-speed oscillation clock

Note Check that $V_{DD} \geq 2.7$ V and set DSCON = 1.

(8) CPU clock changing from high-speed system clock (C) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		MCM0
(C) → (B)	0	10 μs	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

Remark (A) to (K) in Table 5-6 correspond to (A) to (K) in Figure 5-16.

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

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	PORTCTL Register		CMC Register[Note]		CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS2	FXTOUTEN	EXCLKS	OSCSELS	XTSTOP		CSS
(C) → (D) (XT1 clock)	0	1	1	1	0	Necessary	1
(C) → (D) (External subsystem clock)	1	1	1	1	0	Unnecessary	1

Unnecessary if the CPU is operating with the subsystem clock

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

(10) CPU clock changing from subsystem clock (D) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	CKC Register	
	HIOSTOP	MCM0	CSS
(D) → (B)	0	0	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

Unnecessary if this register is already set

Remark (A) to (K) in Table 5-6 correspond to (A) to (K) in Figure 5-16.

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

(11) 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	OSMC Register	OSTC Register	CKC Register	
		MSTOP	FSEL		MCM0	CSS
(D) → (C) (X1 clock: 2 MHz ≤ fx ≤ 10 MHz)	**Note 1**	0	0	Must be checked	1	0
(D) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	**Note 1**	0	1 Note 2	Must be checked	1	0
(D) → (C) (external main clock)	**Note 1**	0	0/1	Must not be checked	1	0

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

Unnecessary if these registers are already set

Notes 1. Set the oscillation stabilization time as follows.

- Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS

2. FSEL = 1 when f_{CLK} > 10 MHz

If a divided clock is selected and f_{CLK} ≤ 10 MHz, use with FSEL = 0 is possible even if f_X > 10 MHz.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 32 ELECTRICAL SPECIFICATIONS).

(12) CPU clock changing from 20 MHz internal high-speed oscillation clock (J) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	DSCCTL Register	
	SELDSC	DSCON
(J) → (B)	0	0

Remark (A) to (K) in Table 5-6 correspond to (A) to (K) in Figure 5-16.

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

(13) • HALT mode (E) set while CPU is operating with internal high-speed oscillation clock (B)
• HALT mode (F) set while CPU is operating with high-speed system clock (C)
• HALT mode (G) set while CPU is operating with subsystem clock (D)
• HALT mode (K) set while CPU is operating with 20 MHz internal high-speed oscillation clock (J)

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

(14) • STOP mode (H) set while CPU is operating with internal high-speed oscillation clock (B)
• STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence) ⟶

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

Remark (A) to (K) in Table 5-6 correspond to (A) to (K) in Figure 5-16.

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

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

Table 5-7. Changing CPU Clock (1/3)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Internal high-speed oscillation clock	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	Operating current can be reduced by stopping internal high-speed oscillator (HIOSTOP = 1).
	External main system clock	Enabling input of external clock from EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	XT1 clock	Stabilization of XT1 oscillation • EXCLKS2 = 0, FXTOUTEN = 1, EXCLKS = 1, OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	
	External subsystem clock	Enabling input of external clock from EXCLKS1 pin • EXCLKS2 = 1, FXTOUTEN = 1, EXCLKS = 1, OSCSELS = 1, XTSTOP = 0	
	20 MHz internal high-speed oscillation clock	Stabilization of DSC oscillation with 20 MHz set by using the option byte • $V_{DD} \geq 2.7$ V • After elapse of oscillation stabilization time (100 μs) after setting to DSCON = 1 • SELDSC = 1	–
X1 clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator • HIOSTOP = 0	X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	XT1 clock	Stabilization of XT1 oscillation • EXCLKS2 = 0, FXTOUTEN = 1, EXCLKS = 1, OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1).
	External subsystem clock	Enabling input of external clock from EXCLKS1 pin • EXCLKS2 = 1, FXTOUTEN = 1, EXCLKS = 1, OSCSELS = 1, XTSTOP = 0	
	20 MHz internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–

Table 5-7. Changing CPU Clock (2/3)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
External main system clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator • HIOSTOP=0	External main system clock input can be disabled (MSTOP = 1).
	X1 clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	XT1 clock	Stabilization of XT1 oscillation • EXCLKS2 = 0, FXTOUTEN = 1, EXCLKS = 1, OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1).
	External subsystem clock	Enabling input of external clock from EXCLKS1 pin • EXCLKS2 = 1, FXTOUTEN = 1, EXCLKS = 1, OSCSELS = 1, XTSTOP = 0	
	20 MHz internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
XT1 clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator and selection of internal high-speed oscillation clock as main system clock • HIOSTOP = 0, MCS = 0	External clock from EXCLKS0 pin can be disabled (XTSTOP = 1).
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	External subsystem clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	20 MHz internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–

Table 5-7. Changing CPU Clock (3/3)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
External subsystem clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator and selection of internal high-speed oscillation clock as main system clock • HIOSTOP = 0, MCS = 0	External clock from EXCLKS0 pin can be disabled (XTSTOP = 1).
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	XT1 clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	20 MHz internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
20 MHz internal high-speed oscillation clock	Internal high-speed oscillation clock	• SELDSC = 0 (Set when changing the clock.)	20 MHz internal high-speed oscillation clock can be stopped (DSCON = 0)
	X1 clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
	External main system clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
	XT1 clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
	External subsystem clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–

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

By setting bits 0 to 2, 4, and 6 (MDIV0 to MDIV2, SDIV, MCM0, CSS) of the system clock control register (CKC), the CPU clock can be switched (between the main system clock and the subsystem clock), main system clock can be switched (between the internal high-speed oscillation clock and the high-speed system clock), and the division ratio of the main system clock can be changed.

The actual switchover operation is not performed immediately after rewriting to CKC; operation continues on the pre-switchover clock for several clocks (see Table 5-8 to Table 5-11).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 7 (CLS) of CKC. Whether the main system clock is operating on the high-speed system clock or internal high-speed oscillation clock can be ascertained using bit 5 (MCS) of CKC.

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

Internal high-speed oscillation clock

Table 5-8. Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{MAINC}	←→ (changing the division ratio)	f_{MAINC}	see **Table 5-9**
f_{SUBC}		f_{SUBC}	
f_{IH}	←→	f_{MX}	see **Table 5-10**
f_{MAINC}	←→	f_{SUBC}	see **Table 5-11**

Table 5-9. Maximum Number of Clocks Required in f_{MAINC} ↔f_{MAINC} (changing the division ratio), f_{SUBC} ↔f_{SUBC} (changing the division ratio)

Set Value Before Switchover	Set Value After Switchover	
	Clock A	Clock B
Clock A		1 + f_A/f_B clock
Clock B	1 + f_B/f_A clock	

Table 5-10. Maximum Number of Clocks Required in f_{IH} ↔f_{MX}

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

(**Remarks** are listed on the next page.)

Table 5-11. Maximum Number of Clocks Required in f_{MAINC} ↔f_{SUBC}

Set Value Before Switchover		Set Value After Switchover	
CSS		CSS	
		0 (f_{CLK} = f_{MAINC})	1 (f_{CLK} = f_{SUBC})
0 (f_{CLK} = f_{MAINC})	f_{MAINC}>f_{SUBC}		1 + 2f_{MAINC}/f_{SUBC} clock
1 (f_{CLK} = f_{SUBC})	f_{MAINC}>f_{SUBC}	2 + f_{SUBC}/f_{MAINC} clock	

Remarks 1. The number of clocks listed in Table 5-9 to Table 5-11 is the number of CPU clocks before switchover.

2. Calculate the number of clocks in Table 5-9 to Table 5-11 by removing the decimal portion.

Example When switching the main system clock from the internal high-speed oscillation clock to the high-speed system clock (@ oscillation with f_{IH} = 8 MHz, f_{MX} = 10 MHz)

1 + f_{IH}/f_{MX} = 1 + 8/10 = 1 + 0.8 = 1.8 → 2 clocks

5.6.8 Conditions before clock oscillation is stopped

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

Table 5-12. Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
Internal high-speed oscillation clock	MCS = 1 or CLS = 1 (The CPU is operating on a clock other than the internal high-speed oscillation clock)	HIOSTOP = 1
X1 clock External main system clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock)	MSTOP = 1
XT1 clock External subsystem clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock)	XTSTOP = 1
20 MHz internal high-speed oscillation clock	SELDSC = 0 (The main system clock is operating on a clock other than the 20 MHz internal high-speed oscillation clock.)	DSCON = 0

5.7 Functions of Clock Output Controller

The clock output is a function to output the clock to extended SFR (3rd SFR).

The clock that is selected by the clock output select register 0 (CKS0) is output to extended SFR (3rd SFR).

5.8 Configuration of Clock Output Controller

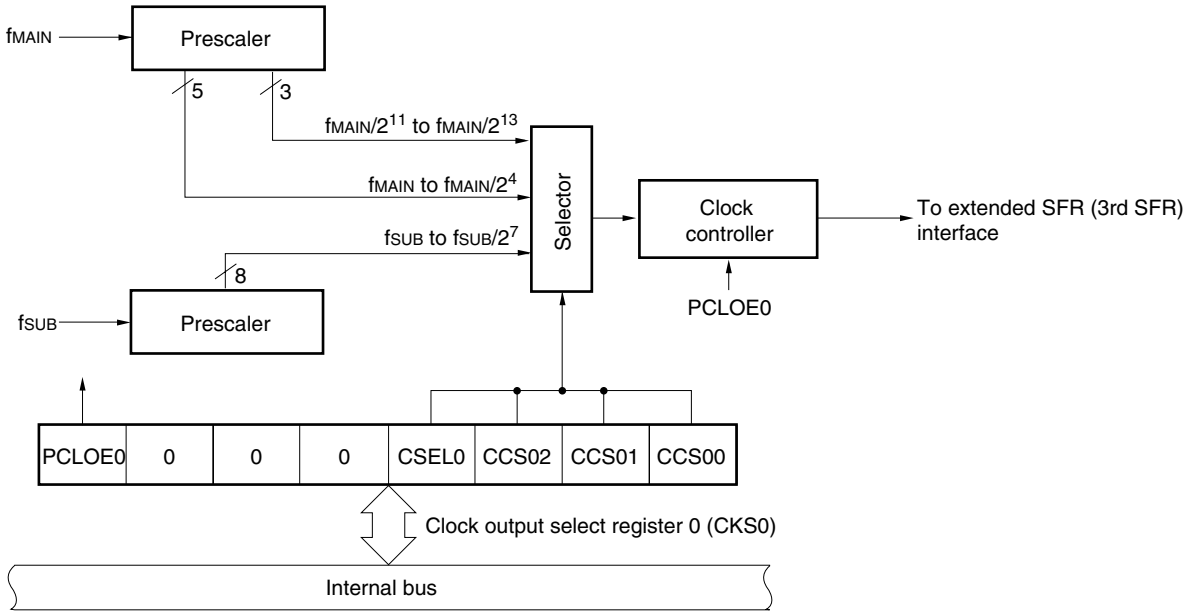
The clock output controller includes the following hardware.

Table 5-13. Configuration of Clock Output Controller

Item	Configuration
Control registers	Clock output select register 0 (CKS0)

Figure 5-17 shows the block diagram of clock output controller.

Figure 5-17. Block Diagram of Clock Output Controller



5.9 Registers Controlling Clock Output Controller

The following register is used to control the clock output controller.

- Clock output select register 0 (CKS0)

(1) Clock output select register 0 (CKS0)

The register set output enable/disable for clock output to the extended SFR (3rd SFR), and set the output clock.

CKS0 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 5-18. Format of Clock Output Select Register 0 (CKS0)

Address: FFFA5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOE0	0	0	0	CSEL0	CCS02	CCS01	CCS00

PCLOE0	Output enable/disable for clock output to the extended SFR (3rd SFR)
0	Output disable to the extended SFR (3rd SFR)
1	Output enable to the extended SFR (3rd SFR)

CSEL0	CCS02	CCS01	CCS00	Selection of the clock output to the extended SFR (3rd SFR)			
					f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz
0	0	0	0	f_{MAIN}	5 MHz	10 MHz[Note]	Setting prohibited[Note]
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz[Note]
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.76 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz
1	0	0	0	f_{SUB}	32.768 kHz		
1	0	0	1	$f_{SUB}/2$	16.384 kHz		
1	0	1	0	$f_{SUB}/2^2$	8.192 kHz		
1	0	1	1	$f_{SUB}/2^3$	4.096 kHz		
1	1	0	0	$f_{SUB}/2^4$	2.048 kHz		
1	1	0	1	$f_{SUB}/2^5$	1.024 kHz		
1	1	1	0	$f_{SUB}/2^6$	512 Hz		
1	1	1	1	$f_{SUB}/2^7$	256 Hz		

Note For the extended SFR (3rd SFR), setting an output clock up to 10 MHz is permitted when 2.7 V ≤ V_{DD}. Setting a clock exceeding 5 MHz at V_{DD} < 2.7 V is prohibited.

(**Cautions** and **Remarks** are listed on the next page.)

Cautions 1. Change the output clock to the extended SFR (3rd SFR) after disabling clock output (PCLOE0 = 0).

2. If the selected clock (f_{MAIN} or f_{SUB}) stops during clock output (PCLOE0 = 1), the output becomes undefined.

3. To shift to STOP mode when the main system clock is selected (CSEL0 = 0), set PCLOE0 = 0 before executing the STOP instruction. When the subsystem clock is selected (CSEL0 = 1), PCLOE0 = 1 can be set because the clock can be output in STOP mode.

4. When 24-bit ΔΣ-type A/D converter is used, output 10 MHz to the extended SFR (3rd SFR).

<R> **5. Transfer clock frequency for the extended SFR (3rd SFR) interface must be 1/4 or less of the clock frequency supplied to the interface, and must satisfy the following.**

- **When $V_{DD} = LV_{DD} \geq 2.7$ V : Set transfer clock frequency to 1.25 MHz or less**
- **When $V_{DD} = LV_{DD} < 2.7$ V : Set transfer clock frequency to 555 kHz or less**

Remarks 1. f_{MAIN}: Main system clock frequency

2. f_{SUB}: Subsystem clock frequency

5.10 Operations of Clock Output Controller

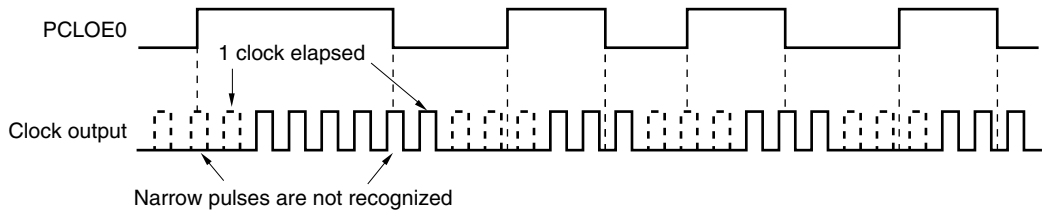
The clock pulse is output as the following procedure.

<1> Select the output frequency to the extended SFR (3rd SFR) with bits 0 to 3 (CCS00 to CCS02, CSEL0) of the clock output select register 0 (CKS0) (output in disabled status).

<2> Set bit 7 (PCLOE0) to 1 to enable clock output to the extended SFR (3rd SFR).

Remark The controller used for outputting the clock to the extended SFR (3rd SFR) starts or stops outputting the clock to the extended SFR (3rd SFR) one clock after enabling or disabling clock output (PCLOE0) is switched. At this time, pulses with a narrow width are not output. Figure 5-19 shows enabling or stopping output using PCLOE0 and the timing of outputting the clock.

Figure 5-19. Remote Control Output Application Example



CHAPTER 6 TIMER ARRAY UNIT

The 78K0R/Lx3-M is provided with two timer array units. Time array unit 0 is provided with eight 16-bit timers and timer array unit 1 is provided with four 16-bit timers. Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more “channels” can be used to create a high-accuracy timer.

The following shows the relationshipt of the function of timer array unit and corresponding channels.

Function			Unit 0								Unit 1			
			0	1	2	3	4	5	6	7	0	1	2	3
Independent Operation Function	Interval timer		√	√	√	√	√	√	√	√	√	√	√	√
	Square wave output		–	–	√	–	–	√	–	√	–	–	–	–
	External event counter		–	–	√	–	√	–	–	√	–	–	–	–
	Divider function		–	–	√	–	–	–	–	–	–	–	–	–
	Input pulse interval measurement		–	–	√	–	√	–	–	√	–	–	–	–
	Measurement of high–/low–level width of input signal		–	–	√	–	√	–	–	√	–	–	–	–
Combination Operation Function	PWM output One-shot pulse output	Master	–	–	–	–	√	–	√	–	–	–	–	–
		Slave	–	–	–	–	–	√	–	√	–	–	–	–

Caution For the combination of master channel and slave channel of the combination operation function, refer to 6.6.2 Basic rules of combination operation function.

Channel 7 of timer array unit 0 can be used to realize LIN-bus reception processing in combination with UART3 of serial array unit 1.

6.1 Functions of Timer Array Unit

The timer array unit has the following functions.

6.1.1 Functions of each channel when it operates independently

Independent operation functions are those functions that can be used for any channel regardless of the operation mode of the other channel (for details, refer to **6.6.1 Overview of single-operation function and combination operation function**).

(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 INTTMpq is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOpq).

(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 (TIpq) has reached a specific value.

(4) Divider function

A clock input from a timer input pin (TI02) is divided and output from an output pin (TO02).

(5) Input pulse interval measurement

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

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

Counting is started by a single edge of the signal input to the timer input pin (TIpq), 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.

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer I/O pins)
mn = 00 to 07, 10 to 13,
pq = 02, 04, 07 (channel that provided with timer input pin)
pq = 02, 05, 07 (channel that provided with timer output pin)

6.1.2 Functions of each channel when it operates with another channel

Combination operation functions are those functions that are attained by using the master channel (mostly the reference timer that controls cycles) and the slave channels (timers that operate following the master channel) in combination (for details, refer to **6.6.1 Overview of single-operation function and combination operation function**).

(1) PWM (Pulse Width Modulator) output

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

(2) One-shot pulse output

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

6.1.3 LIN-bus supporting function (channel 7 of timer array unit 0 only)

(1) Detection of wakeup signal

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD3) of UART3 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 sync break field

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD3) of UART3 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 sync break field.

(3) Measurement of pulse width of sync field

After a sync break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD3) of UART3 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

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 counter register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TIpq pin, RxD3 pin (for LIN-bus)
Timer output	TOpq pins, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register m (TPSm) • Timer channel enable status register m (TEm) • Timer channel start register m (TSm) • Timer channel stop register m (TTm) • Timer input select registers 0, 1 (TIS0, TIS1) • Timer output enable register p (TOEp) • Timer output register p (TOp) • Timer output level register p (TOLp) • Timer output mode register p (TOMp)
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register pq (TSRpq) • Input switch control register (ISC) (channel 7 of timer array unit 0 only) • Noise filter enable register 1 (NFEN1) • Port mode registers 1, 3, 5 (PM1, PM3, PM5) • Port registers 1, 3, 5 (P1, P3, P5)

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer I/O pins)
mn = 00 to 07, 10 to 13,
p = 0, pq = 02, 04, 07 (channel that provided with timer input pin)
p = 0, pq = 02, 05, 07 (channel that provided with timer output pin)

Figures 6-1 and 6-2 show block diagrams.

Figure 6-1. Block Diagram of Timer Array Unit 0

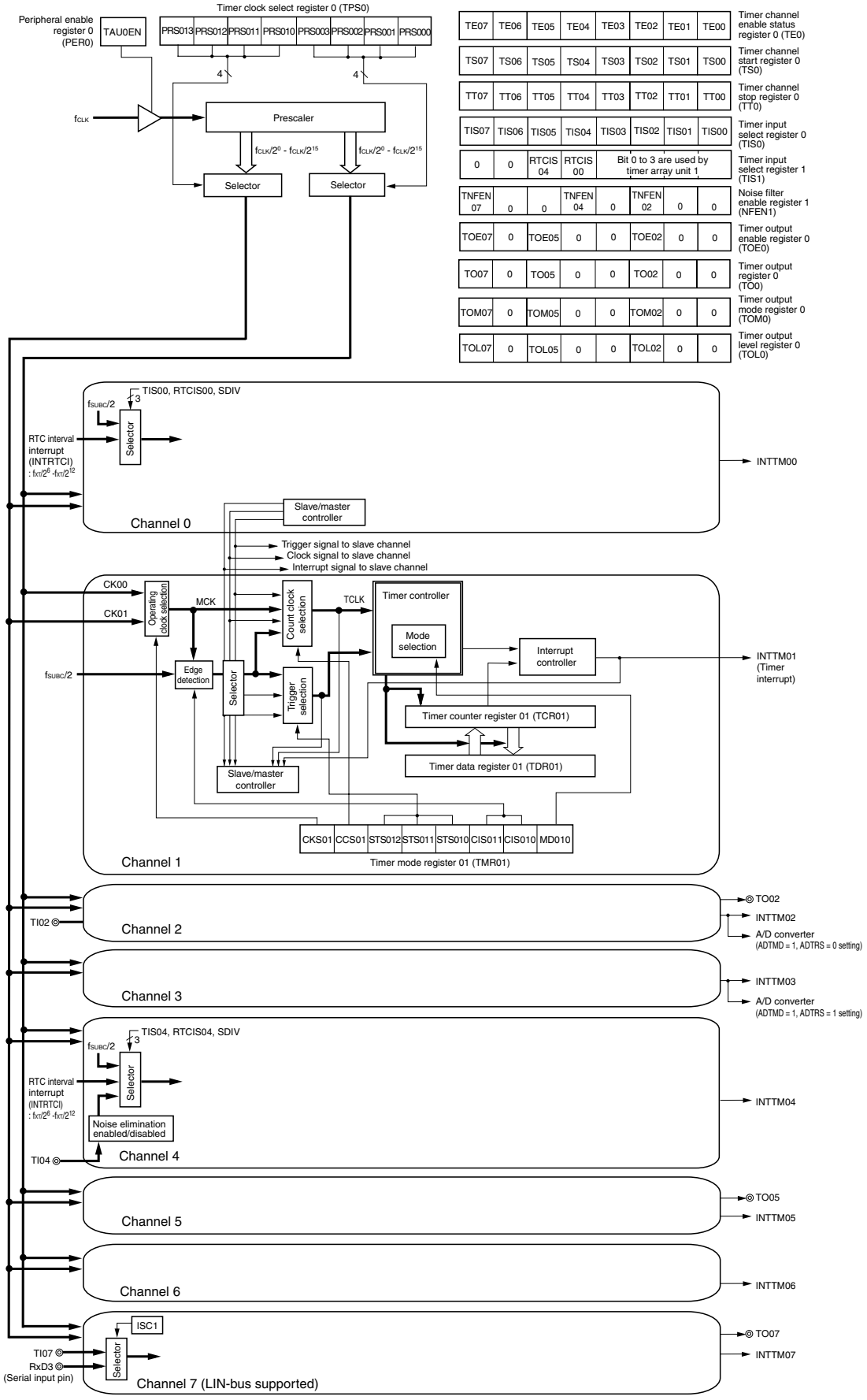
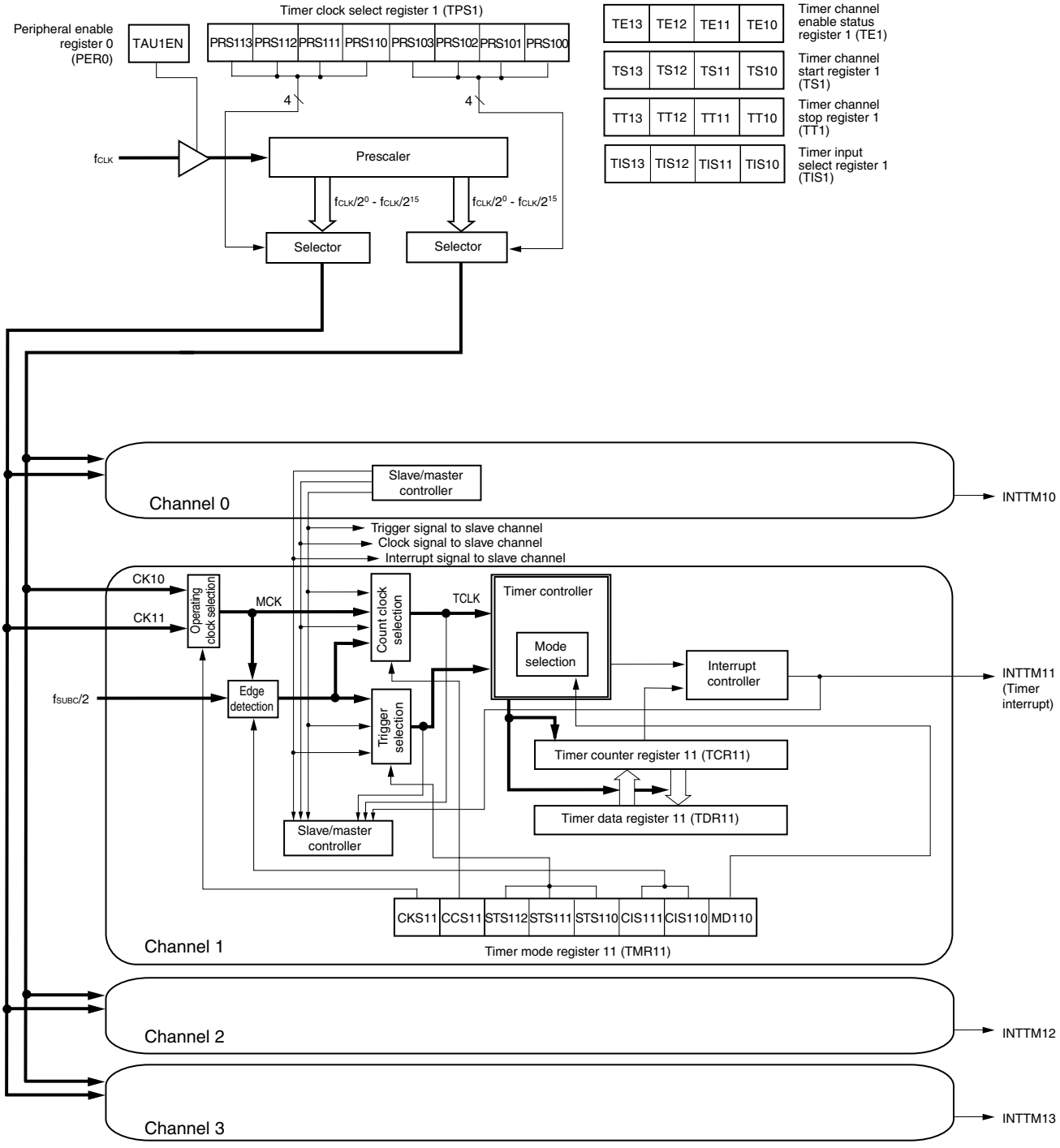


Figure 6-2. Block Diagram of Timer Array Unit 1



(1) Timer/counter register mn (TCRmn)

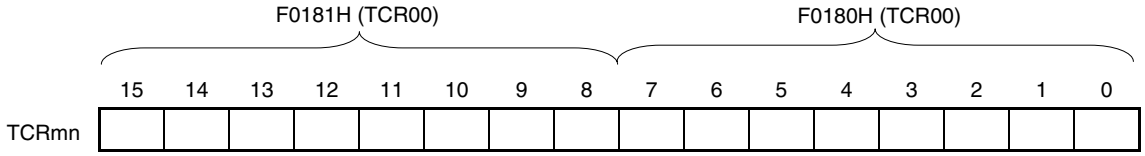
TCRmn is a 16-bit read-only register and is used to count clocks.

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 TMRmn.

Figure 6-3. Format of Timer/Counter Register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07) After reset: FFFFH R
F01C0H, F01C1H (TCR10) to F01C6H, F01C7H (TCR13)



The count value can be read by reading TCRmn.

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

- When the reset signal is generated
- When the TAU0EN bit (TAU0) and TAU1EN bit (TAU1) of peripheral enable register 0 (PER0) is cleared

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
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode

Caution The count value is not captured to TDRmn even when TCRmn is read.

Remark mn: Unit number + Channel number
mn = 00 to 07, 10 to 13

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

Table 6-2. TCRmn Register Read Value in Various Operation Modes

Operation Mode	Count Mode	TCRmn Register Read Value[Note]			
		Operation mode change after reset	Operation mode change after count operation paused (TTmn = 1)	Operation restart after count operation paused (TTmn = 1)	During start trigger wait status after one count
Interval timer mode	Count down	FFFFH	Undefined	Stop value	–
Capture mode	Count up	0000H	Undefined	Stop value	–
Event counter mode	Count down	FFFFH	Undefined	Stop value	–
One-count mode	Count down	FFFFH	Undefined	Stop value	FFFFH
Capture & one-count mode	Count up	0000H	Undefined	Stop value	Capture value of TDRmn register + 1

Note The read values of the TCRmn register when TSmn has been set to "1" while TEmn = 0 are shown. The read value is held in the TCRmn register until the count operation starts.

Remark mn: Unit number + Channel number
mn = 00 to 07, 10 to 13

(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 TMRmn.

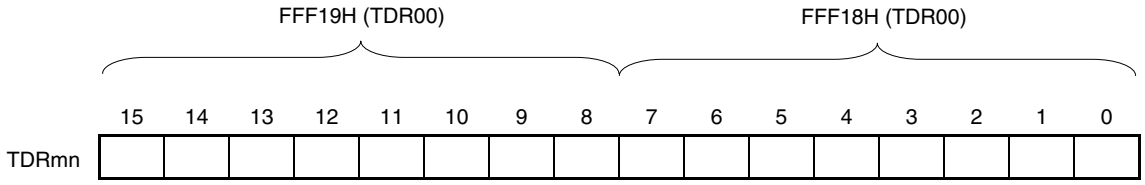
The value of TDRmn can be changed at any time.

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

Reset signal generation clears this register to 0000H.

Figure 6-4. Format of Timer Data Register mn (TDRmn)

Address: FFF18H, FFF19H (TDR00), FFF1AH, FFF1BH (TDR01), FFF64H, FFF65H (TDR02) to FFF6EH, FFF6FH (TDR07) FFF70H, FFF71H (TDR10) to FFF76H, FFF77H (TDR13) After reset: 0000H R/W



(i) When TDRmn is used as compare register

Counting down is started from the value set to TDRmn. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. TDRmn holds its value until it is rewritten.

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

(ii) When TDRpq is used as capture register

The count value of TCRpq is captured to TDRpq when the capture trigger is input.

A valid edge of the TIpq pin can be selected as the capture trigger. This selection is made by TMRpq.

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer I/O pins)
mn = 00 to 07, 10 to 13,
pq = 02, 04, 07 (channel that provided with timer input pin)
pq = 02, 05, 07 (channel that provided with timer output pin)

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register pq (TSRpq)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer input select registers 0, 1 (TIS0, TIS1)
- Timer output enable register p (TOEp)
- Timer output register p (TOp)
- Timer output level register p (TOLp)
- Timer output mode register p (TOMp)
- Input switch control register (ISC)
- Noise filter enable register 1 (NFEN1)
- Port mode registers 1, 3, 5 (PM1, PM3, PM5)
- Port registers 1, 3, 5 (P1, P3, P5)

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer I/O pins)
mn = 00 to 07, 10 to 13,
p = 0, pq = 02, 04, 07 (channel that provided with timer input pin)
p = 0, pq = 02, 05, 07 (channel that provided with timer output pin)

(1) Peripheral enable register 0 (PER0)

PER0 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 timer array unit 0 is used, be sure to set bit 0 (TAU0EN) of this register to 1.

When the timer array unit 1 is used, be sure to set bit 1 (TAU1EN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Caution When setting the timer array unit, be sure to set TAUmEN to 1 first. If TAUmEN = 0, writing to a control register of the timer array unit is ignored, and all read values are default values.

Figure 6-5. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

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

(2) Timer clock select register m (TPSm)

TPSm 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 TPSm, and CKm0 is selected by bits 3 to 0.

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

Rewriting of PRSm00 to PRSm03 bits: Possible only when all the channels set to CKSmn = 0 are in the operation stopped state (TEmn = 0)

Rewriting of PRSm10 to PRSm13 bits: Possible only when all the channels set to CKSmn = 1 are in the operation stopped state (TEmn = 0)

TPSm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TPSm can be set with an 8-bit memory manipulation instruction with TPSmL.

Reset signal generation clears this register to 0000H.

Remark mn: Unit number + Channel number

m = 0, 1, mn = 00 to 07, 10 to 13

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

Address: F01B6H, F01B7H (TPS0) After reset: 0000H R/W

F01DEH, F01DFH (TPS1)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	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	Selection of operation clock (CKmk) Notes 1,2				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz
0	1	1	1	$f_{CLK}/2^7$	15.62 kHz	39.1 kHz	78.1 kHz	156.2 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.76 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	976 Hz	2.44 kHz	4.88 kHz	9.76 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop the timer array unit (TT0 = 00FFH, TT1 = 000FH).

2. Only in the case of SDIV=0, CCSmn=1 and TISmn=1, continuously use of TAUm is allowed, even when changing CPU clock. However, the following limitation is existing.

- When changing CPU clock, source clock decrease/increase occurs as follows.

  Main clock → Subsystem clock (CSS = 0 → 1): −1 clock

  Subsystem clock → Main clock (CSS = 1 → 0): +1 clock

Caution Be sure to clear bits 15 to 8 to "0".

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

2. k = 0, 1

3. mn: Unit number + Channel number

m = 0, 1, mn = 00 to 07, 10 to 13

(3) Timer mode register mn (TMRmn)

TMRmn sets an operation mode of channel n of timer array unit m. It is used to select an operation clock (MCK), a count clock, whether the timer operates as the master or a slave, a start trigger and a capture trigger, the valid edge of the timer input, and an operation mode (interval, capture, event counter, one-count, or capture & one-count).

Rewriting TMRmn is prohibited when the register is in operation (when TEm = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEm = 1) (for details, see **6.7 Operation of Timer Array Unit as Independent Channel** and **6.8 Operation of Plural Channels of Timer Array Unit**).

TMRmn can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6-7. Format of Timer Mode Register mn (TMRmn) (1/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR10) to F01CEH, F01CFH (TMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKS mn	0	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

CKS mn	Selection of operation clock (MCK) of channel n
0	Operation clock CKm0 set by TPSm register
1	Operation clock CKm1 set by TPSm register
Operation clock MCK is used by the edge detector. A count clock (TCLK) and a sampling clock are generated depending on the setting of the CCSmn bit.	

CCS mn	Selection of count clock (TCLK) of channel n
0	Operation clock MCK specified by CKSmn bit
1	Valid edge of input signal input from TIpq pin, $f_{SUB}/2$, $f_{SUB}/4$, or INTRTCI (the timer input used with channel x is selected by using TISm register).
Count clock TCLK is used for the timer/counter, output controller, and interrupt controller. If CCSmn = 1, use the count clock under the following condition. • The frequency of the operating clock selected by using CKSmn ≥ The frequency of the clock selected by using TISmn × 2	

Caution TMR02, TMR04, TMR07 registers: Be sure to clear bits 14, 13, 5, and 4 to "0".
TMR00, TMR01, TMR03, TMR05, TMR06, TMR10 to TMR13 registers: Be sure to clear bits 14, 13, 11, and 5 to 1 to "0".

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer I/O pins)
mn = 00 to 07, 10 to 13,
pq = 02, 04, 07 (channel that provided with timer input pin)

Figure 6-7. Format of Timer Mode Register mn (TMRmn) (2/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR10) to F01CEH, F01CFH (TMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn	0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

MASTERmn	Selection of slave/master of channel n
0	Operates as slave channel with combination operation function.
1	Operates as master channel with combination operation function.
Only the even channel can be set as a master channel (MASTERmn = 1). Be sure to use the odd channel as a slave channel (MASTERmn = 0). Clear MASTERmn to 0 for a channel that is used with the independent operation function.	

STSmn2	STSmn1	STSmn0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of TIpq pin input signal, $f_{SUB}/2$, $f_{SUB}/4$, or INTRTCI is used as both the start trigger and capture trigger.
0	1	0	Both the edges of TIpq pin input signal, $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 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 combination operation function).
Other than above			Setting prohibited

CISmn1	CISmn0	Selection of valid edge of TIpq pin input signal , $f_{SUB}/2$, $f_{SUB}/4$, or INTRTCI (the timer input used with channel x is selected by using TISm register).
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.		

Caution TMR02, TMR04, TMR07 registers: Be sure to clear bits 14, 13, 5, and 4 to "0".
TMR00, TMR01, TMR03, TMR05, TMR06, TMR10 to TMR13 registers: Be sure to clear bits 14, 13, 11, and 5 to 1 to "0".

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer I/O pins)
mn = 00 to 07, 10 to 13,
pq = 02, 04, 07 (channel that provided with timer input pin)

Figure 6-7. Format of Timer Mode Register mn (TMRmn) (3/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR10) to F01CEH, F01CFH (TMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKS mn	0	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

MD mn3	MD mn2	MD mn1	MD mn0	Operation mode of channel n	Count operation of TCR	Independent operation
0	0	0	1/0	Interval timer mode	Counting down	Possible
0	1	0	1/0	Capture mode	Counting up	Possible
0	1	1	0	Event counter mode	Counting down	Possible
1	0	0	1/0	One-count mode	Counting down	Impossible
1	1	0	0	Capture & one-count mode	Counting up	Possible
Other than above				Setting prohibited		
The operation of MDmn0 bits varies depending on each operation mode (see following table).						

Cautions 1. TMR02, TMR04, TMR07 registers: Be sure to clear bits 14, 13, 5, and 4 to "0".
TMR00, TMR01, TMR03, TMR05, TMR06, TMR10 to TMR13 registers: Be sure to clear bits 14, 13, 11, and 5 to 1 to "0".
2. Channel 0, 1, 3, and 6 of timer array unit 0 and channel 0 to 3 of timer array unit 1 can be set only to the interval mode.

Remark mn: Unit number + Channel number
mn = 00 to 07, 10 to 13

Figure 6-7. Format of Timer Mode Register mn (TMRmn) (4/4)

Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above))	MD mn0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• Capture mode (0, 1, 0)	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
	1	Start trigger is valid during counting operation[Note]. At that time, interrupt is also generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated, either.
Other than above		Setting prohibited

Note If the start trigger (TSmn = 1) is issued during operation, the counter is cleared, an interrupt is generated, and recounting is started.

Remark mn: Unit number + Channel number
mn = 00 to 07, 10 to 13

(4) Timer status register pq (TSRpq)

TSRpq indicates the overflow status of the counter of channel n.

TSRpq is valid only in the capture mode (MDpq3 to MDpq1 = 010B) and capture & one-count mode (MDpq3 to MDpq1 = 110B). It will not be set in any other mode. See Table 6-3 for the operation of the OVFpq bit in each operation mode and set/clear conditions.

TSRpq can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TSRpq can be set with an 8-bit memory manipulation instruction with TSRpqL.

Reset signal generation clears this register to 0000H.

Figure 6-8. Format of Timer Status Register pq (TSRpq)

Address: F01A4H, F01A5H (TSR02), F01A8H, F01A9H (TSR04) After reset: 0000H R
F01AEH, F01AFH (TSR07)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSRpq	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVFpq

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

Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
pq = 02, 04, 07

Table 6-3. OVFpq Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVFpq	Set/clear conditions
• Capture mode	clear	When no overflow has occurred upon capturing
• Capture & one-count mode	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	–
	set	(Use prohibited, not set/cleared)

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

(5) Timer channel enable status register m (TEm)

TEm is used to enable or stop the timer operation of each channel.

When a bit of timer channel start register m (TSm) is set to 1, the corresponding bit of this register is set to 1.

When a bit of timer channel stop register m (TTm) is set to 1, the corresponding bit of this register is cleared to 0.

TEm can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TEm can be set with a 1-bit or 8-bit memory manipulation instruction with TEmL.

Reset signal generation clears this register to 0000H.

Figure 6-9. Format of Timer Channel Enable Status Register m (TEm)

Address: F01B0H, F01B1H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	0	0	0	0	TE07	TE06	TE05	TE04	TE03	TE02	TE01	TE00

Address: F01D8H, F01D9H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE1	0	0	0	0	0	0	0	0	0	0	0	0	TE13	TE12	TE11	TE10

TE mn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.

Remark mn: Unit number + Channel number
m = 0, 1, mn = 00 to 07, 10 to 13

(6) Timer channel start register m (TSm)

TSm is a trigger register that is used to clear a timer counter (TCRmn) and start the counting operation of each channel.

When a bit (TSmn) of this register is set to 1, the corresponding bit (TEmn) of timer channel enable status register m (TEm) is set to 1. TSmn is a trigger bit and cleared immediately when TEmn = 1.

TSm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TSm can be set with a 1-bit or 8-bit memory manipulation instruction with TSmL.

Reset signal generation clears this register to 0000H.

Figure 6-10. Format of Timer Channel Start Register m (TSm)

Address: F01B2H, F01B3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	0	0	0	0	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

Address: F01DAH, F01DBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1	0	0	0	0	0	0	0	0	0	0	0	0	TS13	TS12	TS11	TS10

TSmn	Operation enable (start) trigger of channel n
0	No trigger operation
1	TEmn is set to 1 and the count operation becomes enabled. The TCRmn count operation start in the count operation enabled state varies depending on each operation mode (see Table 6-4).

Caution Be sure to clear bits 15 to 8 of TS0 and bits 15 to 4 of TS1 to "0".

Remarks 1. When the TSm register is read, 0 is always read.

2. mn: Unit number + Channel number

m = 0, 1, mn = 00 to 07, 10 to 13

Table 6-4. Operations from Count Operation Enabled State to TCRmn Count Start

Timer operation mode	Operation when TSmn = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TSmn=1) until count clock generation. The first count clock loads the value of TDRmn to TCRmn and the subsequent count clock performs count down operation (see **6.3 (6) (a) Start timing in interval timer mode**).
• Event counter mode	Writing 1 to TSmn bit loads the value of TDRmn to TCRmn. The subsequent count clock performs count down operation. The external trigger detection selected by STSmn2 to STSmn0 bits in the TMRmn register does not start count operation (see **6.3 (6) (b) Start timing in event counter mode**).
• Capture mode	No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to TCRmn and the subsequent count clock performs count up operation (see **6.3 (6) (c) Start timing in capture mode**).
• One-count mode	When TEmn = 0, writing 1 to TSmn bit sets the start trigger wait state. No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of TDRmn to TCRmn and the subsequent count clock performs count down operation (see **6.3 (6) (d) Start timing in one-count mode**).
• Capture & one-count mode	When TEmn = 0, writing 1 to TSmn bit sets the start trigger wait state. No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to TCRmn and the subsequent count clock performs count up operation (see **6.3 (6) (e) Start timing in capture & one-count mode**).

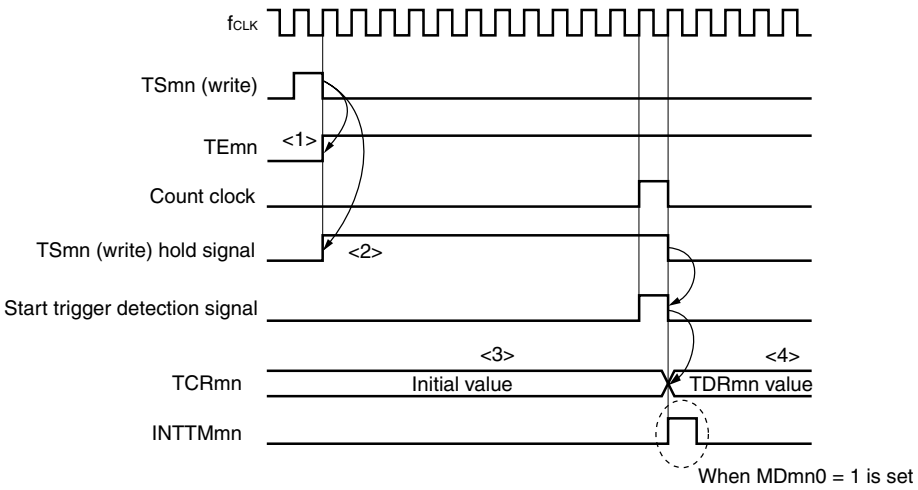
Caution Channels 0, 1, 3, and 6 of timer array unit 0 and channels 0 to 3 of timer array unit 1 can be set only to the interval mode.

(a) Start timing in interval timer mode

<1> Writing 1 to TSmn sets TEmn = 1
<2> The write data to TSmn is held until count clock generation.
<3> TCRmn holds the initial value until count clock generation.
<4> On generation of count clock, the “TDRmn value” is loaded to TCRmn and count starts.

Remark mn: Unit number + Channel number
mn = 00 to 07, 10 to 13

Figure 6-11. Start Timing (In Interval Timer Mode)



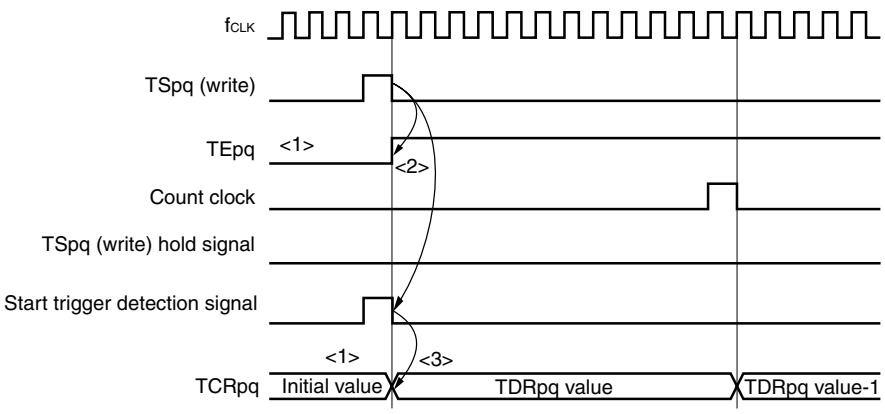
Caution In the first cycle operation of count clock after writing TSmn, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

Remark mn: Unit number + Channel number
mn = 00 to 07, 10 to 13

(b) Start timing in event counter mode

<1> While TEpq is set to 0, TCRpq holds the initial value.
<2> Writing 1 to TSpq sets 1 to TEpq.
<3> As soon as 1 has been written to TSpq and 1 has been set to TEpq, the "TDRpq value" is loaded to TCRpq to start counting.
<4> After that, the TCRpq value is counted down according to the count clock.

Figure 6-12. Start Timing (In Event Counter Mode)

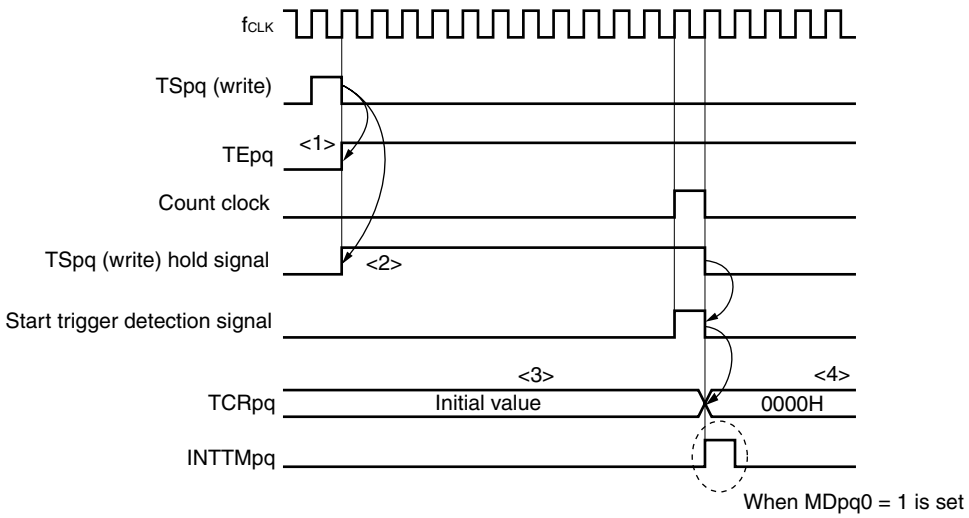


Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
pq = 02, 04, 07

(c) Start timing in capture mode

<1> Writing 1 to TSpq sets TEpq = 1
<2> The write data to TSpq is held until count clock generation.
<3> TCRpq holds the initial value until count clock generation.
<4> On generation of count clock, 0000H is loaded to TCRpq and count starts.

Figure 6-13. Start Timing (In Capture Mode)



Caution In the first cycle operation of count clock after writing TSpq, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDpq0 = 1.

Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
pq = 02, 04, 07

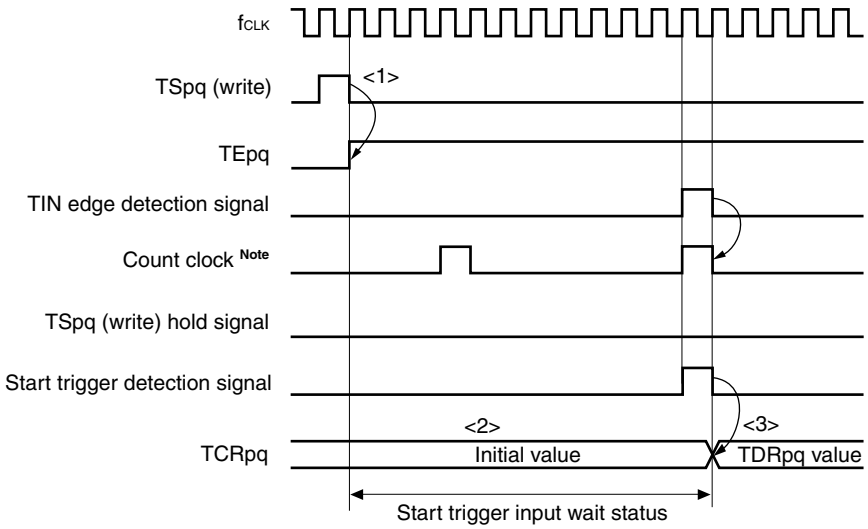
(d) Start timing in one-count mode

<1> Writing 1 to TSpq sets TEpq = 1

<2> Enters the start trigger input wait status, and TCRpq holds the initial value.

<3> On start trigger detection, the "TDRpq value" is loaded to TCRpq and count starts.

Figure 6-14. Start Timing (In One-count Mode)



Note When the one-count mode is set, the operation clock (MCK) is selected as count clock (CCSpq = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TIpq is used).

Remark pq: Unit number + Channel number (only for channels provided with timer output pins)
pq = 02, 05, 07

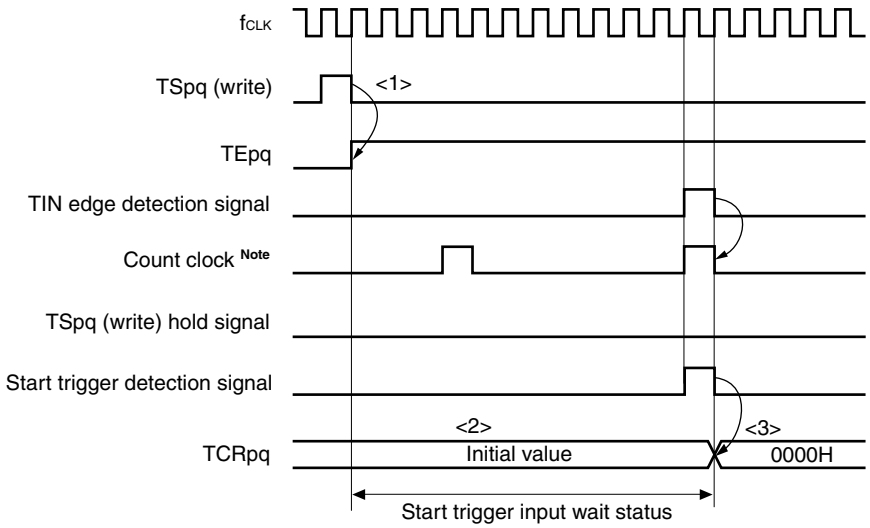
(e) Start timing in capture & one-count mode

<1> Writing 1 to TSpq sets TEpq = 1

<2> Enters the start trigger input wait status, and TCRpq holds the initial value.

<3> On start trigger detection, 0000H is loaded to TCRpq and count starts.

Figure 6-15. Start Timing (In Capture & One-count Mode)



Note When the capture & one-count mode is set, the operation clock (MCK) is selected as count clock (CCSpq = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when Tlpq is used).

(7) Timer channel stop register m (TTm)

TTm is a trigger register that is used to clear a timer counter (TCRmn) and start the counting operation of each channel.

When a bit (TTmn) of this register is set to 1, the corresponding bit (TEmn) of timer channel enable status register 0 (TEm) is cleared to 0. TTmn is a trigger bit and cleared to 0 immediately when TEmn = 0.

TTm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TTm can be set with a 1-bit or 8-bit memory manipulation instruction with TTmL.

Reset signal generation clears this register to 0000H.

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer I/O pins)
mn = 00 to 07, 10 to 13,
pq = 02, 04, 07 (channel that provided with timer input pin)
pq = 02, 05, 07 (channel that provided with timer output pin)

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
TT0	0	0	0	0	0	0	0	0	TT07	TT06	TT05	TT04	TT03	TT02	TT01	TT00

Address: F01DCH, F01DDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT1	0	0	0	0	0	0	0	0	0	0	0	0	TT13	TT12	TT11	TT10

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	Operation is stopped (stop trigger is generated).

Caution Be sure to clear bits 15 to 8 of TT0 and bits 15 to 4 of TT1 to "0".

Remarks 1. When the TTm register is read, 0 is always read.

2. mn: Unit number + Channel number
m = 0, 1, mn = 00 to 07, 10 to 13

(8) Timer input select registers 0, 1 (TIS0, TIS1)

TIS0 and TIS1 use can be set to the input signal of a timer input pin (TIpq), half the frequency of the subsystem clock ($f_{SUB}/2$), one fourth the frequency of the subsystem clock ($f_{SUB}/4$), or an RTC interval interrupt (INTRTCI) as the timer input. The timer input can be selected for each channel.

TIS0 and TIS1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
mn = 00 to 07, 10 to 13,
pq = 02, 04, 07 (channel that provided with timer input pin)

Figure 6-17. Format of Timer Input Select Registers 0, 1 (TIS0, TIS1) (1/2)

Address: FFF3EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	TIS07	TIS06	TIS05	TIS04	TIS03	TIS02	TIS01	TIS00

Address: FFF3FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS1	0	0	RTCIS04	RTCIS00	TIS13	TIS12	TIS11	TIS10

Figure 6-17. Format of Timer Input Select Registers 0, 1 (TIS0, TIS1) (2/2)

• Channels 1 to 3 and 5 to 7 of timer array unit 0 and channels 0 to 3 of timer array unit 1

TISmn	SDIV	Selection of Timer input used with channel (mn = 01 to 03, 05 to 07)[Note1]
0	×	Set enable only for TIS02 and TIS07 Input signal of timer input pin (TI02, TI07)
1	0	$f_{SUB}/2$
	1	$f_{SUB}/4$

• Channels 0 and 4 of timer array unit 0

TISmn	RTCISmn	SDIV	Selection of Timer input used with channel (mn = 00, 04)[Note2]
0	×	×	Set enable only for TIS04 Input signal of timer input pin (TI04)
1	0	0	$f_{SUB}/2$
		1	$f_{SUB}/4$
	1	0	RTC Interval interrupt (INTRTCI)
		1	Setting prohibited

Notes 1. Since TIS01, TIS03, TIS05, TIS06, and TIS10 to TIS13 do not have timer input pins, only $f_{SUB}/2$ and $f_{SUB}/4$ (TISmn = 1 (mn = 01, 03, 05, 06, 10 to 13)) can be selected.

2. Since TIS00 does not have timer input pin, only $f_{SUB}/2$, $f_{SUB}/4$, and RTC interval interrupt (INTRTCI) (TIS00 = 1) can be selected.

Caution When the LIN-bus communication function is used, select the input signal of the RxD3 pin by setting ISC1 to 1 and TIS07 = 0.

Remarks 1. pq: Unit number + Channel number (only for channels provided with timer input pins)
pq = 02, 04, 07

2. ×: don't care

3. f_{SUB}: Subsystem select clock

4. SDIV: Bit 3 of the system clock control register (CKC)

(9) Timer output enable register p (TOEp)

TOEp is used to enable or disable timer output of each channel.

Channel q for which timer output has been enabled becomes unable to rewrite the value of the TOpq bit of the timer output register (TOp) 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 (TOpq).

TOEp can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOEp can be set with a 1-bit or 8-bit memory manipulation instruction with TOEpL.

Reset signal generation clears this register to 0000H.

Figure 6-18. Format of Timer Output Enable Register p (TOEp)

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
TOE0	0	0	0	0	0	0	0	0	TOE07	0	TOE05	0	0	TOE02	0	0

TOEpq	Timer output enable/disable of channel q
0	The TOpq operation stopped by count operation (timer channel output bit). Writing to the TOpq bit is enabled. The TOpq pin functions as data output, and it outputs the level set to the TOpq bit. The output level of the TOpq pin can be manipulated by software.
1	The TOpq operation enabled by count operation (timer channel output bit). Writing to the TOpq bit is disabled (writing is ignored). The TOpq pin functions as timer output, and the TOEpq is set or reset depending on the timer operation. The TOpq pin outputs the square-wave or PWM depending on the timer operation.

Caution Be sure to clear bits 15 to 8, 6, 4, 3, 1, and 0 to "0".

Remark pq: Unit number + Channel number (only for channels provided with timer output pins)
p = 0, pq = 02, 05, 07

(10) Timer output register p (TOp)

TOp 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 (TOpq) of each channel.

This register can be rewritten by software only when timer output is disabled (TOEpq = 0). When timer output is enabled (TOEpq = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P12/TO02/SO20/TxD2, P16/TO05, or P33/TO07/TI07/INTP3 pin as a port function pin, set the corresponding TOpq bit to "0".

TOp can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOp can be set with an 8-bit memory manipulation instruction with TOpL.

Reset signal generation clears this register to 0000H.

Figure 6-19. Format of Timer Output Register p (TOp)

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
TO0	0	0	0	0	0	0	0	0	TO07	0	TO05	0	0	TO02	0	0

TOpq	Timer output of channel q
0	Timer output value is "0".
1	Timer output value is "1".

Caution Be sure to clear bits 15 to 8, 6, 4, 3, 1, and 0 to "0".

Remark pq: Unit number + Channel number (only for channels provided with timer output pins)
p = 0, pq = 02, 05, 07

(11) Timer output level register p (TOLp)

TOLp is a register that controls the timer output level of each channel.

The setting of the inverted output of channel q by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOEpq = 1) in the combination operation mode (TOMpq = 1). In the toggle mode (TOMpq = 0), this register setting is invalid.

TOLp can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOLp can be set with an 8-bit memory manipulation instruction with TOLpL.

Reset signal generation clears this register to 0000H.

Figure 6-20. Format of Timer Output Level Register p (TOLp)

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
TOL0	0	0	0	0	0	0	0	0	TOL07	0	TOL05	0	0	TOL02	0	0

TOLpq	Control of timer output level of channel q
0	Positive logic output (active-high)
1	Inverted output (active-low)

Caution Be sure to clear bits 15 to 8, 6, 4, 3, 1, and 0 to "0".

Remarks 1. If the value of this register is rewritten during timer operation, the timer output is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.

2. pq: Unit number + Channel number (only for channels provided with timer output pins)
p = 0, pq = 02, 05, 07

(12) Timer output mode register p (TOMp)

TOMp is used to control the timer output mode of each channel.

When a channel is used for the independent operation function, set the corresponding bit of the channel to be used to 0.

When a channel is used for the combination operation function (PWM output or one-shot pulse 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 q by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOEpq = 1).

TOMp can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOMp can be set with an 8-bit memory manipulation instruction with TOMpL.

Reset signal generation clears this register to 0000H.

Figure 6-21. Format of Timer Output Mode Register p (TOMp)

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
TOM0	0	0	0	0	0	0	0	0	TOM07	0	TOM05	0	0	TOM02	0	0

TOMpq	Control of timer output mode of channel q
0	Toggle mode (to produce toggle output by timer interrupt request signal (INTTMmn))
1	Combination operation mode (set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTMpq) of the slave channel)

Caution Be sure to clear bits 15 to 8, 6, 4, 3, 1, and 0 to "0".

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer output pins)

mn = 00 to 07, 10 to 13, (when master channel: mn = 04, 06)

p = 0, pq = 02, 05, 07

(13) Input switch control register (ISC)

ISC is used to implement LIN-bus communication operation with channel 7 of timer array unit 0 in association with serial array unit 1.

When bit 1 of this register is set to 1, the input signal of the serial data input pin (RxD3) is selected as a timer input signal.

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

Reset signal generation clears this register to 00H.

Figure 6-22. Format of Input Switch Control Register (ISC)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	ISC4	ISC3	ISC2	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 RxD3 pin is used as timer input (wakeup signal detection).

Caution Be sure to clear bits 5 to 7 to "0".

Remarks 1. When the LIN-bus communication function is used, select the input signal of the RxD3 pin by setting ISC1 to 1 and TIS07 to 0.

2. Bits 0 and 2 to 4 of ISC are not used with timer array unit.

(14) Noise filter enable register 1 (NFEN1)

NFEN1 is used to set whether the noise filter can be used for the timer input signal of timer array unit 0 to each channel.

Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal.

When the noise filter is ON, match detection and synchronization of the 2 clocks is performed with the CPU/peripheral hardware clock (f_{CLK}). When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock (f_{CLK}).

NFEN1 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 Noise Filter Enable Register 1 (NFEN1)

Address: F0061H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	0	0	TNFEN04	0	TNFEN02	0	0

TNFEN07	Enable/disable using noise filter of TI07/P33/TO07/INTP3 pin input signal or RxD0/P81/INTP9 pin input signal[Note]
0	Noise filter OFF
1	Noise filter ON

TNFEN04	Enable/disable using noise filter of TI04/P53/SEG36 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02/P52/SEG37 pin input signal
0	Noise filter OFF
1	Noise filter ON

Note The applicable pin can be switched by setting ISC1 of the ISC register.

ISC1 = 0: Whether or not to use the noise filter of TI07 pin can be selected.
ISC1 = 1: Whether or not to use the noise filter of RxD3 pin can be selected.

(15) Port mode registers 1, 3, 5 (PM1, PM3, PM5)

These registers set input/output of ports 1, 3, and 5 in 1-bit units.

When using the P12/TO02/SO20/TxD2, P16/TO05, and P33/TO07/TI07/INTP3 pins for timer output, set PM12, PM16, and PM33 and the output latches of P12, P16, and P33 to 0.

When using the P52/TI02/SEG37, P53/TI04/SEG36, and P33/TO07/TI07/INTP3 pins for timer input, set PM52, PM53, and PM33 to 1. At this time, the output latches of P52, P53, and P33 may be 0 or 1.

PM1, PM3, and PM5 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 6-24. Format of Port Mode Registers 1, 3, 5 (PM1, PM3, PM5)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30

Address: FFF25H After reset: FFH R/W

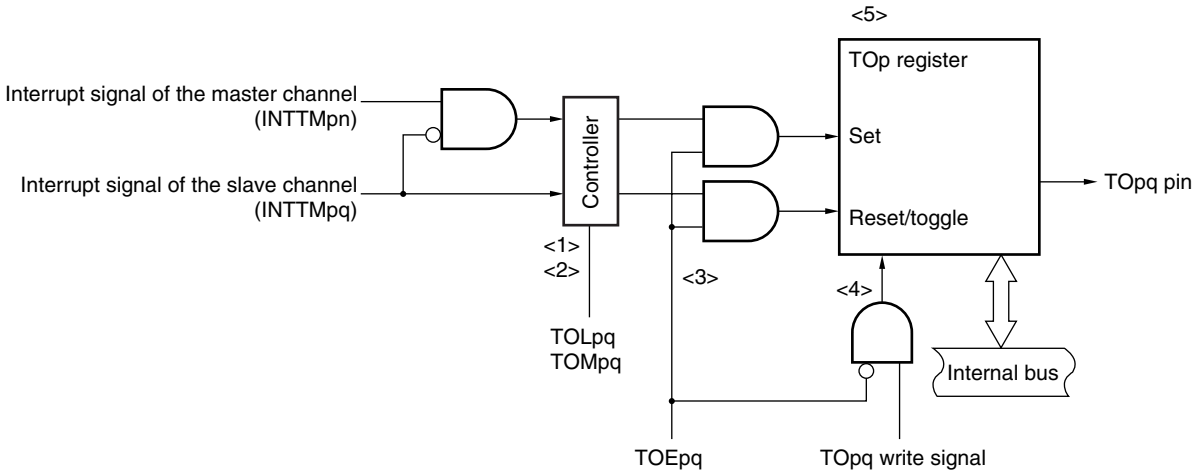
Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

PMmn	Pmn pin I/O mode selection (m = 1, 3, 5; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 Channel Output (TOpq pin) Control

6.4.1 TOpq pin output circuit configuration

Figure 6-25. Output Circuit Configuration



The following describes the TOpq pin output circuit.

<1> When TOMpq = 0 (toggle mode), the set value of the TOLp register is ignored and only INTTMpq (slave channel timer interrupt) is transmitted to the TOp register.

<2> When TOMpn, TOMpq = 1 (combination operation mode), both INTTMpn (master channel timer interrupt) and INTTMpq (slave channel timer interrupt) are transmitted to the TOp register.
At this time, the TOLp register becomes valid and the signals are controlled as follows:

When TOLpq = 0: Forward operation (INTTMpn → set, INTTMpq → reset)
When TOLpq = 1: Reverse operation (INTTMpn → reset, INTTMpq → set)

When INTTMpn and INTTMpq are simultaneously generated, (0% output of PWM), INTTMpq (reset signal) takes priority, and INTTMpn (set signal) is masked.

<3> When TOEpq = 1, INTTMpn (master channel timer interrupt) and INTTMpq (slave channel timer interrupt) are transmitted to the TOpq register. Writing to the TOp register (TOpq write signal) becomes invalid.
When TOEpq = 1, the TOpq pin output never changes with signals other than interrupt signals.
To initialize the TOpq pin output level, it is necessary to set TOEpq = 0 and to write a value to TOpq.

<4> When TOEpq = 0, writing to TOpq bit to the target channel (TOpq signal) becomes valid. When TOEpq = 0, neither INTTMpn (master channel timer interrupt) nor INTTMpq (slave channel timer interrupt) is transmitted to TOpq register.

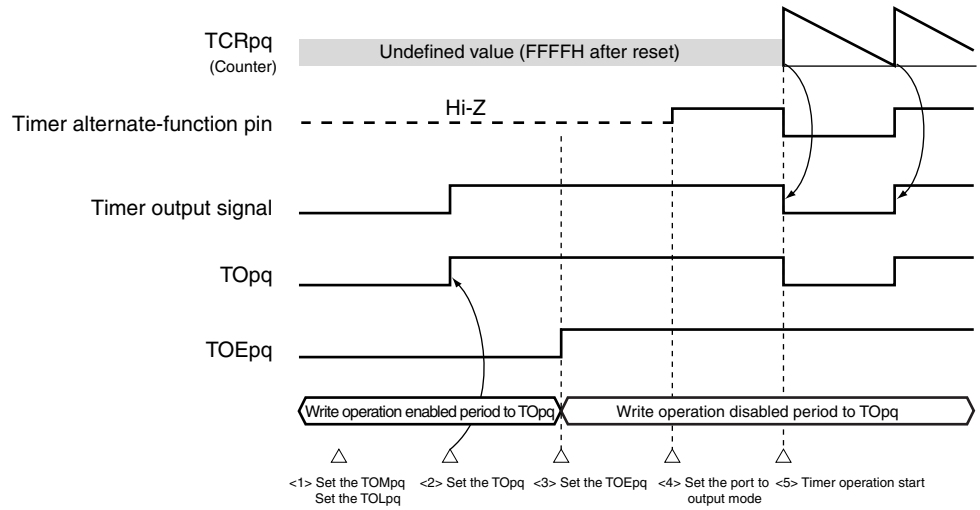
<5> The TOp register can always be read, and the TOpq pin output level can be checked.

Remark pq: Unit number + Channel number (only for channels provided with timer output pins)
p = 0, pq = 02, 05, 07
n: master channel number (n = 4, 6)

6.4.2 TOpq Pin Output Setting

The following figure shows the procedure and status transition of TOpq out put pin from initial setting to timer operation start.

Figure 6-26. Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOMpq bit (0: Toggle mode, 1: Combination operation mode)
- TOLpq bit (0: Forward output, 1: Reverse output)

<2> The timer output signal is set to the initial status by setting TOpq.
<3> The timer output operation is enabled by writing 1 to TOEpq (writing to TOpq is disabled).
<4> The port I/O setting is set to output (see **6.3 (15) Port mode registers 1, 3, 5 (PM1, PM3 and PM5)**).
<5> The timer operation is enabled (TSpq = 1).

Remark pq: Unit number + Channel number (only for channels provided with timer output pins)
pq = 02, 05, 07

6.4.3 Cautions on Channel Output Operation

(1) Changing values set in registers TOp,TOEp,TOLp, and TOMp during timer operation

Since the timer operations (operations of TCRpq and TDRpq) are independent of the TOpq output circuit and changing the values set in TOp, TOEp, TOLp, and TOMp does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TOpq pin by timer operation, however, set TOp, TOEp, TOLp, and TOMp to the values stated in the register setting example of each operation.

When the values set in TOEp, TOLp, and TOMp (except for TOp) are changed close to the timer interrupt (INTTMpq), the waveform output to the TOpq pin may be different depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMpq) signal generation timing.

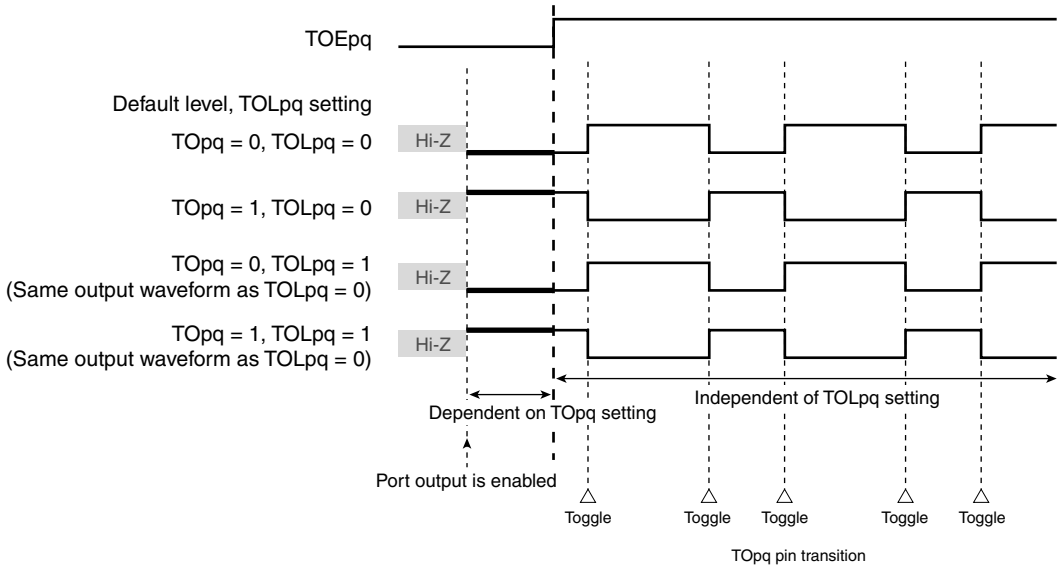
(2) Default level of TOpq pin and output level after timer operation start

The following figure shows the TOpq pin output level transition when writing has been done in the state of TOEpq = 0 before port output is enabled and TOEpq = 1 is set after changing the default level.

(a) When operation starts with TOMpq = 0 setting (toggle output)

The setting of TOLpq is invalid when TOMpq = 0. When the timer operation starts after setting the default level, the toggle signal is generated and the output level of TOpq pin is reversed.

Figure 6-27. TOpq Pin Output Status at Toggle Output (TOMpq = 0)



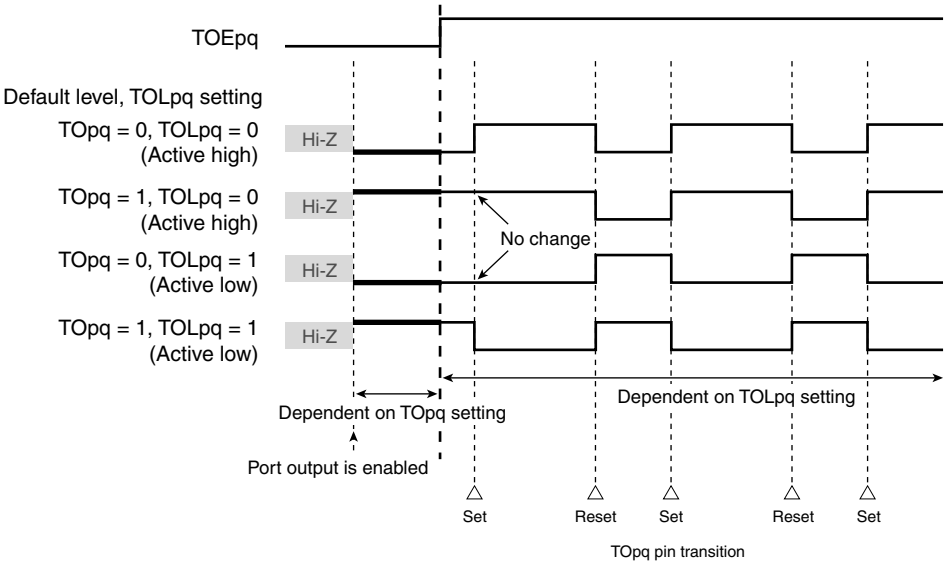
Remarks 1. Toggle: Reverse TOpq pin output status

2. pq: Unit number + Channel number (only for channels provided with timer output pins)
p = 0, pq = 02, 05, 07

(b) When operation starts with TOMpq = 1 setting (Combination operation mode (PWM output))

When TOMpq = 1, the active level is determined by TOLpq setting.

Figure 6-28. TOpq Pin Output Status at PWM Output (TOMpq = 1)



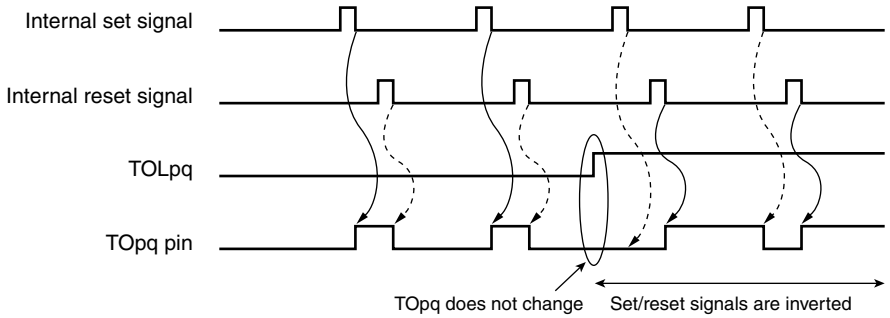
(3) Operation of TOpq pin in combination operation mode (TOMpq = 1)

(a) When TOLpq setting has been changed during timer operation

When the TOLpq setting has been changed during timer operation, the setting becomes valid at the generation timing of TOpq change condition. Rewriting TOLpq does not change the output level of TOpq.

The following figure shows the operation when the value of TOLpq has been changed during timer operation (TOMpq = 1).

Figure 6-29. Operation when TOLpq Has Been Changed during Timer Operation



Remarks 1. Set: The output signal of TOpq pin changes from inactive level to active level.
Reset: The output signal of TOpq pin changes from active level to inactive level.

2. pq: Unit number + Channel number (only for channels provided with timer output pins)
p = 0, pq = 02, 05, 07

(b) Set/reset timing

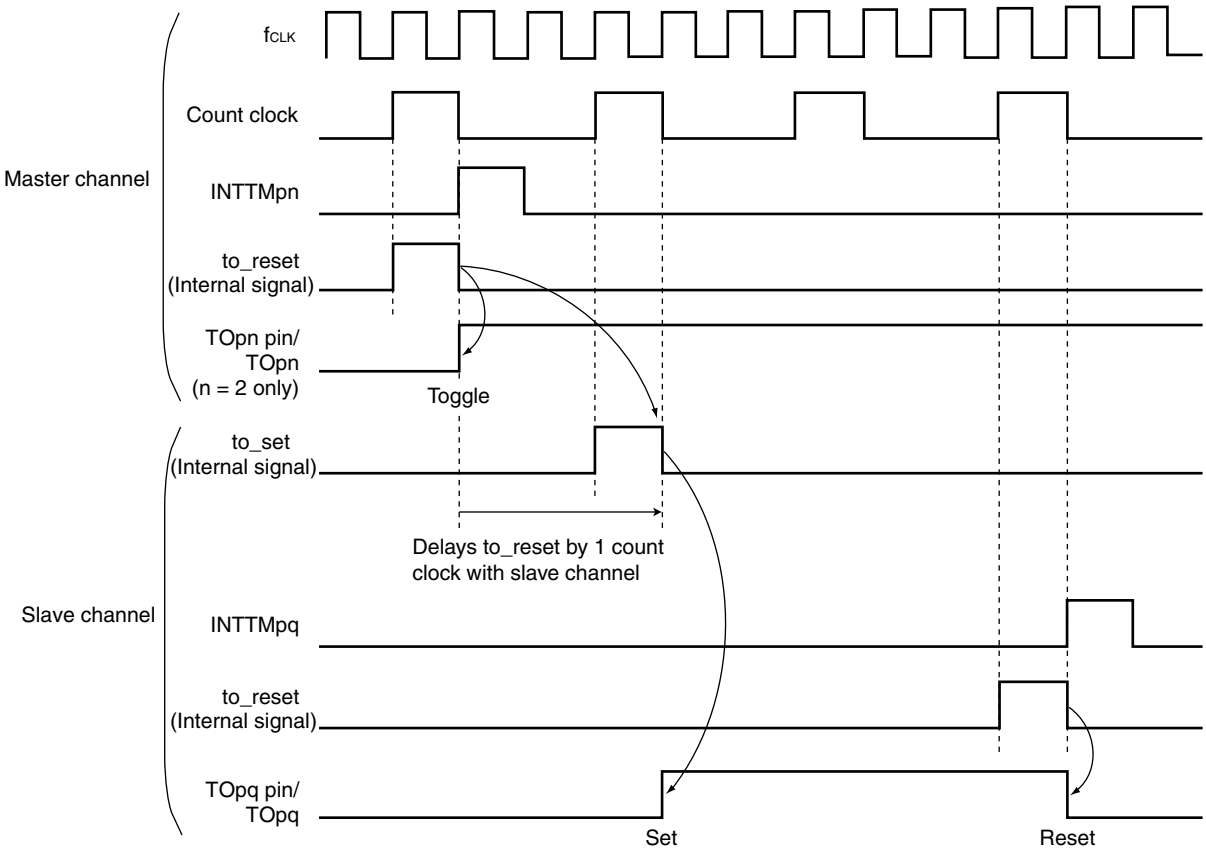
To realize 0%/100% output at PWM output, the TOpn pin/TOpn (n = 2 only) set timing at master channel timer interrupt (INTTMpn) generation is delayed by 1 count clock by the slave channel.

If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-30 shows the set/reset operating statuses where the master/slave channels are set as follows.

- Master channel: TOEpn = 1, TOMpn = 0, TOLpn = 0
- Slave channel: TOEpq = 1, TOMpq = 1, TOLpq = 0

Figure 6-30. Set/Reset Timing Operating Statuses



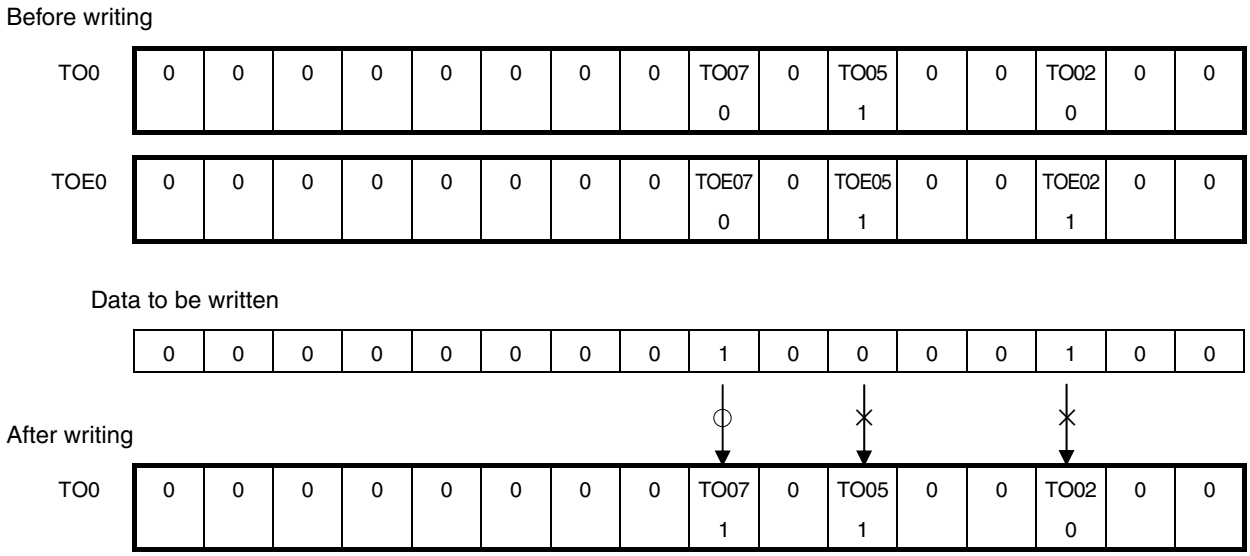
Remarks 1. to_reset: TOpn pin reset/toggle signal
to_set: TOpn pin set signal

2. pq: Unit number + Channel number (only for channels provided with timer output pins)
p = 0, pq = 02, 05, 07
n: master channel number (n = 4, 6)

6.4.4 Collective manipulation of TOpq bits

In the TOp register, the setting bits for all the channels are located in one register in the same way as the TSp register (channel start trigger). Therefore, TOpq of all the channels can be manipulated collectively. Only specific bits can also be manipulated by setting the corresponding TOEpq = 0 to a target TOpq (channel output).

Figure 6-31. Example of TO0q Bits Collective Manipulation



Writing is done only to TOpq bits with TOEpq = 0, and writing to TOpq bits with TOEpq = 1 is ignored.

TOpq (channel output) to which TOEpq = 1 is set is not affected by the write operation. Even if the write operation is done to TOpq, it is ignored and the output change by timer operation is normally done.

Caution When TOEpq = 1, even if the output by timer interrupt of each timer (INTTMpq) contends with writing to TOpq, output is normally done to TOpq pin.

Remark pq: Unit number + Channel number (only for channels provided with timer output pins)
p = 0, pq = 02, 05, 07

6.4.5 Timer Interrupt and TOpq Pin Output at Operation Start

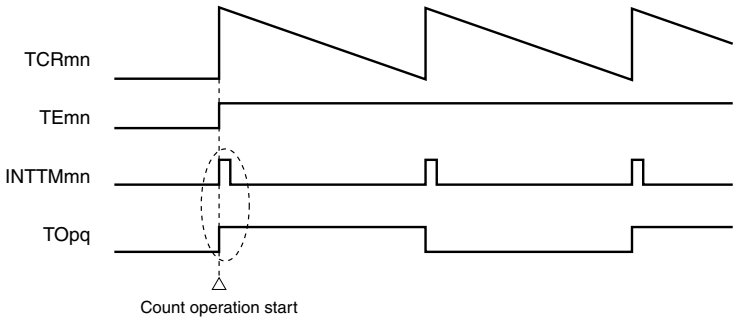
In the interval timer mode or capture mode, the MDmn0 bit in the TMRmn register sets whether or not to generate a timer interrupt at count start.

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 TOpq output is controlled.

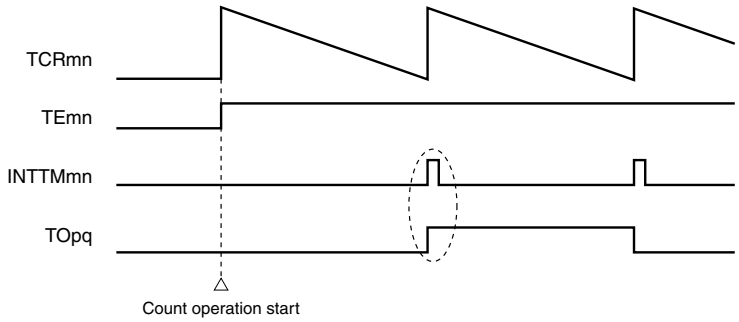
Figures 6-34 and 6-35 show operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 6-32. When MDmn0 is set to 1



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOpq performs a toggle operation.

Figure 6-33. When MDmn0 is set to 0



When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOpq does not change either. After counting one cycle, INTTMmn is output and TOpq performs a toggle operation.

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer outpupt pins)
mn = 00 to 07, 10 to 13, pq = 02, 05, 07

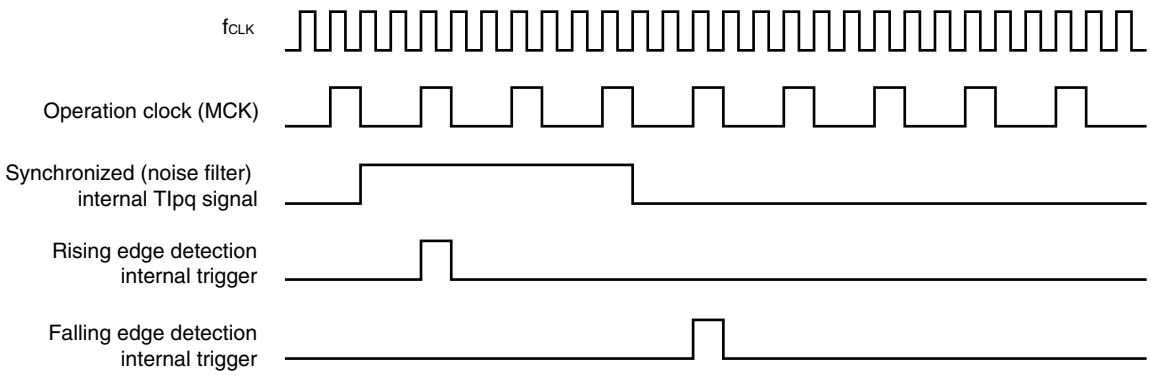
6.5 Channel Input Control

6.5.1 Edge detection circuit

(1) Edge detection basic operation timing

Edge detection circuit sampling is done in accordance with the operation clock (MCK).

Figure 6-34. Edge Detection Basic Operation Timing



Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
pq = 02, 04, 07

6.6 Basic Function of Timer Array Unit

6.6.1 Overview of single-operation function and combination operation function

The timer array unit consists of several channels and has a single-operation function that allows each channel to operate independently, and a combination operation function that uses two or more channels in combination.

The single-operation function can be used for any channel, regardless of the operation mode of the other channels.

The combination operation function is realized by combining a master channel (reference timer that mainly counts periods) and a slave channel (timer that operates in accordance with the master channel), and several rules must be observed when using this function.

Caution Timer array unit 1 cannot use combination operation function.

6.6.2 Basic rules of combination operation function

The basic rules of using the combination operation function are as follows.

(1) Only an even channel (channel 4, channel 6) can be set as a master channel.
(2) Channel 5 and channel 7 can be set as a slave channel.
(3) The slave channel must be lower than the master channel.
Example: If channel 4 of TAU0 is set as a master channel,
channel 5 can be set as a slave channel.
If channel 6 of TAU0 is set as a master channel,
channel 7 can be set as a slave channel.

(4) 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 4 and 6 of TAU0 are set as master channels,
channel 5 can be set as the slave channel of master channel 4. Channel 7 cannot be set as the slave channel of master channel 4.

(5) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKS bit (bit 15 of the TMRpq register) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
(6) A master channel can transmit INTTMpn (interrupt), start software trigger, and count clock to the lower channels.
(7) A slave channel can use the INTTMpn (interrupt), start software trigger, and count clock of the master channel, but it cannot transmit its own INTTMpq (interrupt), start software trigger, and count clock to the lower channel.
(8) A master channel cannot use the INTTMpn (interrupt), start software trigger, and count clock from the higher master channel.
(9) To simultaneously start channels that operate in combination, the TSpn and TSpq bits of the channels in combination must be set at the same time.
(10) During a counting operation, the TSpn and TSpq bits of all channels that operate in combination or only the master channel can be set. Only TSpq of a slave channel cannot be set.
(11) To stop the channels in combination simultaneously, the TTpn and TTpq bits of the channels in combination must be set at the same time.

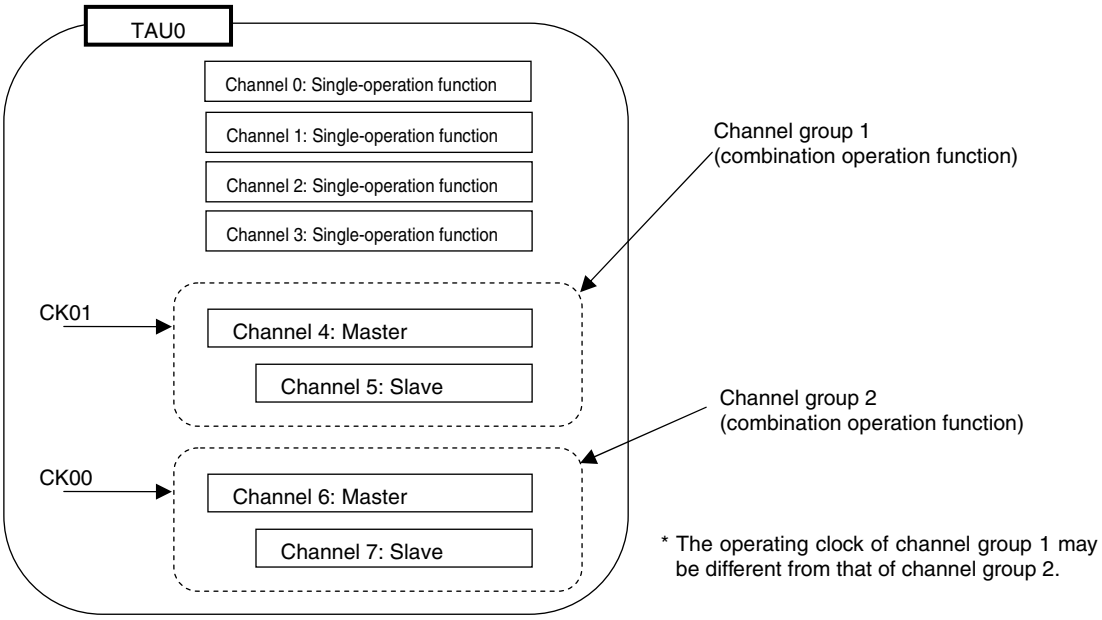
Remark pq: Unit number + Channel number (only for channels provided with timer output pins)
p = 0, pq = 05, 07
n: master channel number (n = 4, 6)

6.6.3 Applicable range of basic rules of combination operation function

The rules of the combination operation function are applied in a channel group (a master channel and slave channels forming one combination operation function).

If two or more channel groups that do not operate in combination are specified, the basic rules of the combination operation function in **6.6.2 Basic rules of combination operation function** do not apply to the channel groups.

Example



6.7 Operation of Timer Array Unit as Independent Channel

6.7.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTMmn (timer interrupt) at fixed intervals. The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

(2) Operation as square wave output

TOpq performs a toggle operation as soon as INTTMpq has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOpq can be calculated by the following expressions.

• Period of square wave output from TOpq = Period of count clock × (Set value of TDRpq + 1) × 2
• Frequency of square wave output from TOpq = Frequency of count clock/{(Set value of TDRpq + 1) × 2}

The valid edge of TIpq pin input signal, the valid edge of $f_{SUB}/2$, the valid edge of $f_{SUB}/4$, or the valid edge of INTRTCI can be selected as the count clock, in addition to CKm0 and CKm1. Consequently, the interval timer can be operated, regardless of the f_{CLK} frequency (main system clock, subsystem clock).

When changing the clock selected as f_{CLK} (changing the value of the system clock control register (CKC)), stop the timer array units 0 and 1 (TAUS0, TAUS1) (TT0 = 00FFH, TT1 = 000FH) first.

Only in the case of SDIV=0, CCSmn=1 and TISmn=1, continuously use of TAUm is allowed, even when changing CPU clock. However, the following limitation is existing.

- When changing CPU clock, source clock decrease/increase occurs as follows.
  Main clock → Subsystem clock (CSS = 0 → 1): −1 clock
  Subsystem clock → Main clock (CSS = 1 → 0): +1 clock

TCRmn operates as a down counter in the interval timer mode.

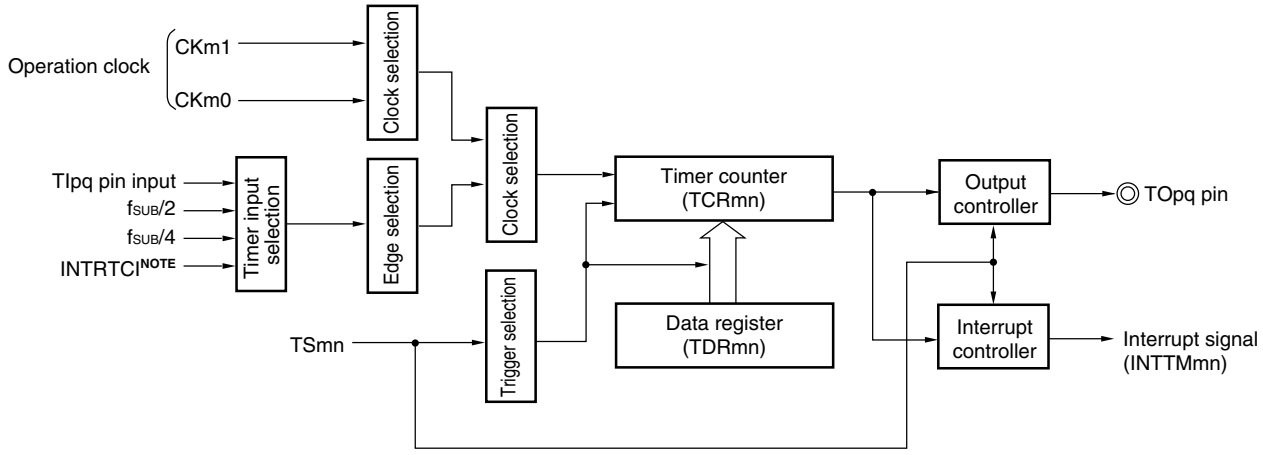
TCRmn loads the value of TDRmn at the first count clock after the channel start trigger bit (TSmn) is set to 1. If MDmn0 of TMRmn = 0 at this time, INTTMmn is not output and TOpq is not toggled. If MDmn0 of TMRmn = 1, INTTMmn is output and TOpq is toggled.

After that, TCRmn count down in synchronization with the count clock.

When TCRmn = 0000H, INTTMmn is output and TOpq is toggled at the next count clock. At the same time, TCRmn loads the value of TDRmn again. After that, the same operation is repeated.

TDRmn can be rewritten at any time. The new value of TDRmn becomes valid from the next period.

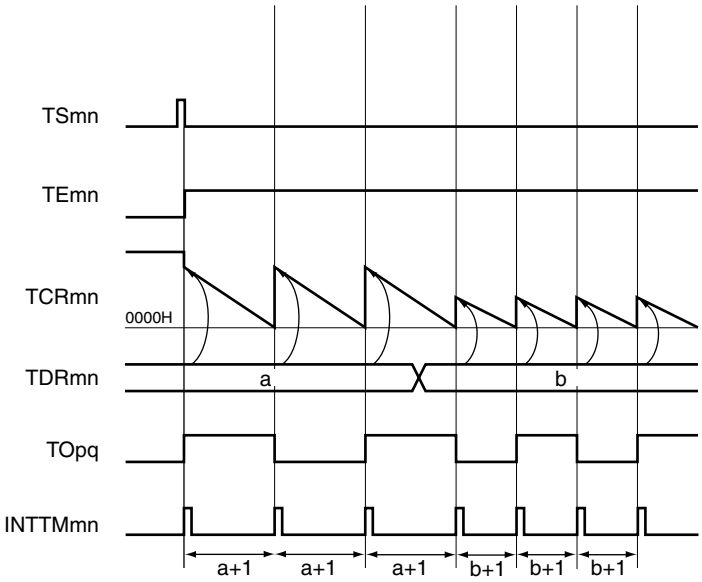
Figure 6-35. Block Diagram of Operation as Interval Timer/Square Wave Output



Note Channels 0 and 4 of timer array unit 0 only

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer output pins)
m = 0, 1, mn = 00 to 07, 10 to 13, pq = 02, 05, 07

Figure 6-36. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)

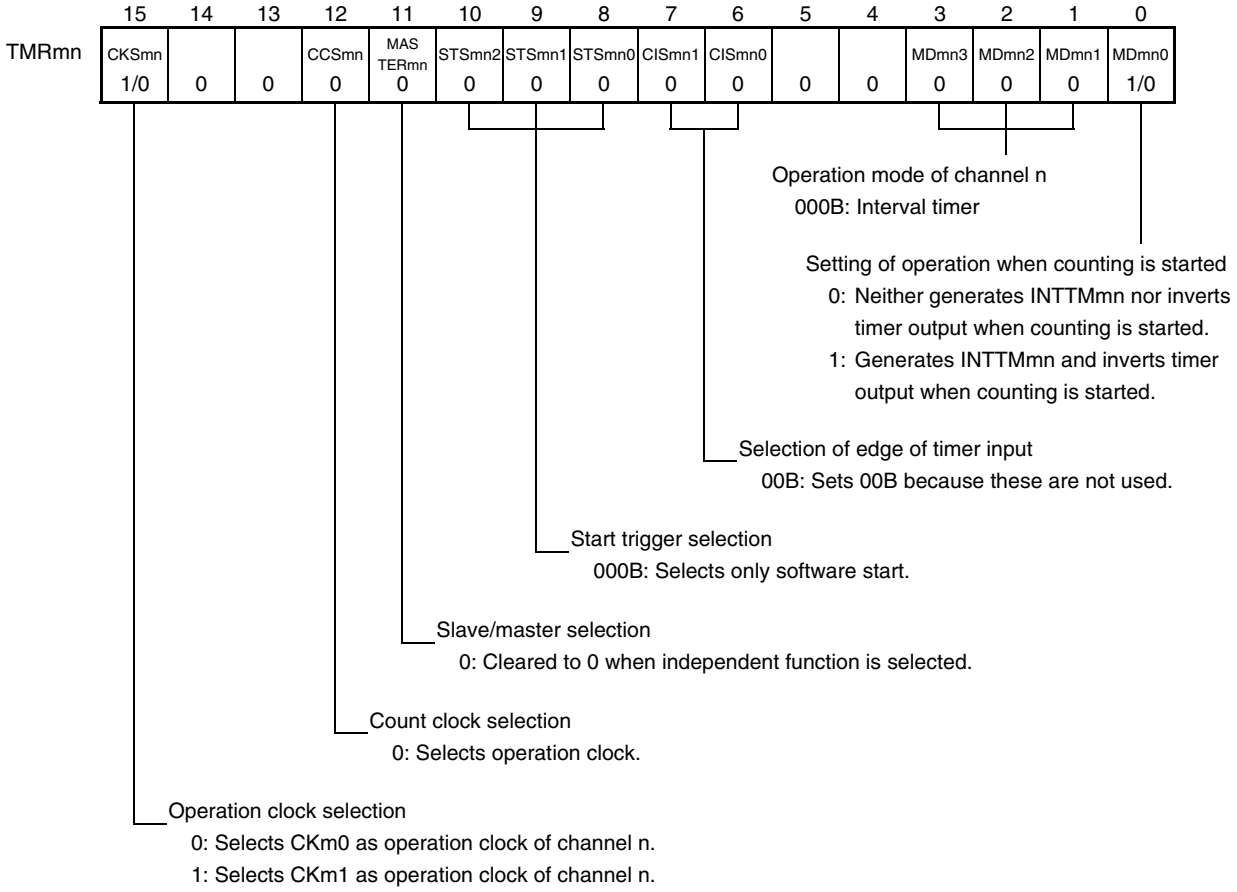


Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer output pins)
mn = 00 to 07, 10 to 13, pq = 02, 05, 07

Figure 6-37. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/3)

(1) When CKm0 or CKm1 is selected as count clock

(a) Timer mode register mn (TMRmn)



(b) Timer output register p (TOp)

	Bit q	
TOp	TOpq 1/0	0: Outputs 0 from TOpq. 1: Outputs 1 from TOpq.

(c) Timer output enable register p (TOEp)

	Bit q	
TOEp	TOEpq 1/0	0: Stops the TOpq output operation by counting operation. 1: Enables the TOpq output operation by counting operation.

(d) Timer output level register p (TOLp)

	Bit q	
TOLp	TOLpq 0	0: Cleared to 0 when TOMpq = 0 (toggle mode)

(e) Timer output mode register p (TOMp)

	Bit q	
TOMp	TOMpq 0	0: Sets toggle mode.

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer output pins)
m = 0, 1, mn = 00 to 07, 10 to 13, pq = 02, 05, 07

Figure 6-37. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/3)

(2) When the timer input (TIpq pin input, f_{SUB}/4, f_{SUB}/2 or INTRTCI) is selected as count clock[Note] (1/2)

Note The timer input is selected by using TISpq bit, SDIV bit, and RTCISpq bit. For details, refer to **Figure 6-17 Format of Timer Input Select Registers 0, 1 (TIS0, TIS1).**

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn			CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	0	0	1/0

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.
1: Generates INTTMmn and inverts timer output when counting is started.

Selection of edge of timer input
00B: detects falling edge.
01B: detects rising edge.
10B: detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when independent function is selected.

Count clock selection
1: Selects timer input valid edge.

Operation clock selection
0: Selects CKm0 as operation clock of channel n.
1: Selects CKm1 as operation clock of channel n.
f_{CLK} (no division) is selected as selected operation clock by TPSm register.

(b) Timer clock select register m (TPSm)

	Bits 7 to 4, 3 to 0
TPSm	PRSmk3 to PRSmk0
	0000

0000B: Selects f_{CLK} (no division) as operation clock selected by CKSmn of TMRmn register.
k = 0 (bits 0 to 3) when CKm0 is selected and k = 1 (bits 4 to 7) when CKm1 is selected

(c) Timer output register p (TOp)

	Bit q
TOp	TOpq
	1/0

0: Outputs 0 from TOpq.
1: Outputs 1 from TOpq.

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer output pins)
m = 0, 1, mn = 00 to 07, 10 to 13, pq = 02, 05, 07

Figure 6-37. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (3/3)

(2) When the timer input (TIpq pin input, f_{SUB}/4, f_{SUB}/2 or INTRTCI) is selected as count clock (2/2)

(e) Timer output enable register p (TOEp)

	Bit q	
TOEp	TOEpq 1/0	0: Stops the TOpq output operation by counting operation. 1: Enables the TOpq output operation by counting operation.

(f) Timer output level register p (TOLp)

	Bit q	
TOLp	TOLpq 0	0: Cleared to 0 when TOMpq = 0 (toggle mode)

(g) Timer output mode register p (TOMp)

	Bit q	
TOMp	TOMpq 0	0: Sets toggle mode.

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer output pins)
m = 0, 1, mn = 00 to 07, 10 to 13, pq = 02, 05, 07

Figure 6-38. 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 TAU0EN or TAU1EN bits of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel). If timer input is selected for the count clock, set the timer input (TIpq pin input, $f_{SUB}/4$, $f_{SUB}/2$, or INTRTCI) by using the TISpq, SDIV, and RTCISpq bits. Sets interval (period) value to the TDRmn register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOpq output Clears the TOMpq bit of the TOMm register to 0 (toggle mode). Clears the TOLpq bit to 0. Sets the TOpq bit and determines default level of the TOpq output. →	The TOmn pin goes into Hi-Z output state. The TOpq default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets TOEpq to 1 and enables operation of TOpq. →	TOpq does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOpq pin outputs the TOpq set level.
Operation start	Sets TOEpq to 1 (only when operation is resumed). 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 TDRmn is loaded to TCRmn at the count clock input. INTTMmn is generated and TOpq performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set values of TMRmn, TOMp, and TOLp registers cannot be changed. Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOp and TOEp registers can be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of TDRmn is loaded to TCRmn 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 bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn holds count value and stops. The TOpq output is not initialized but holds current status.
	TOEpq is cleared to 0 and value is set to TOp register. →	The TOpq pin outputs the TOpq set level.

Operation is resumed.

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer output pins)
m = 0, 1, mn = 00 to 07, 10 to 13, pq = 02, 05, 07

Figure 6-38. Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOpq pin output level Clears TOpq bit to 0 after the value to be held is set to the port register. →	The TOpq pin output level is held by port function.
	When holding the TOpq pin output level is not necessary Switches the port mode register to input mode. →	The TOpq pin output level goes into Hi-Z output state.
	The TAU0EN or TAU1EN bits of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOpq bit is cleared to 0 and the TOpq pin is set to port mode.)

Remark mn: Unit number + Channel number, pq: Unit number + Channel number (only for channels provided with timer output pins)
m = 0, 1, mn = 00 to 07, 10 to 13, pq = 02, 05, 07

6.7.2 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 TIpq pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDRpq + 1

TCRpq operates as a down counter in the event counter mode.

When the channel start trigger bit (TSpq) is set to 1, TCRpq loads the value of TDRpq.

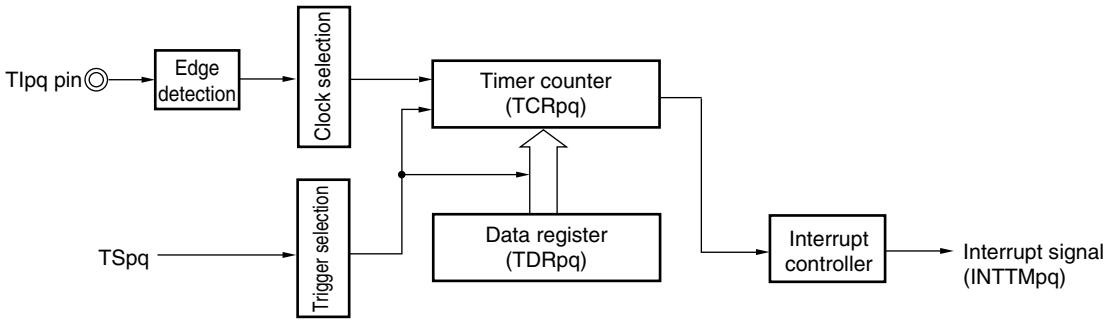
TCRpq counts down each time the valid input edge of the TIpq pin has been detected. When TCRpq = 0000H, TCRpq loads the value of TDRpq again, and outputs INTTMpq.

After that, the above operation is repeated.

TOpq must not be used because its waveform depends on the external event and irregular.

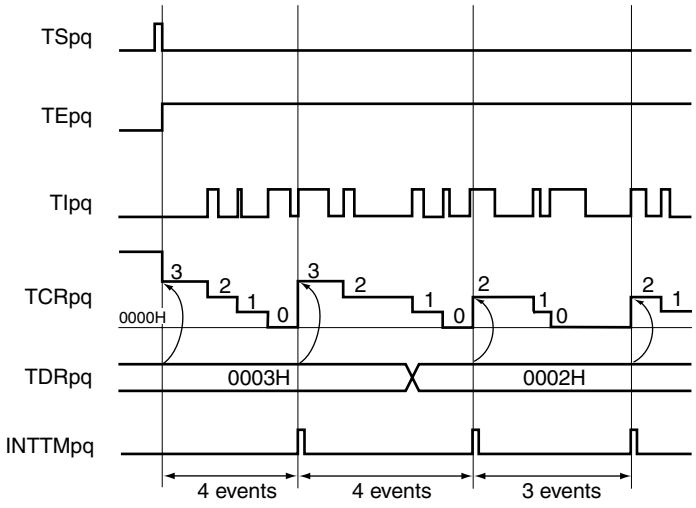
TDRpq can be rewritten at any time. The new value of TDRpq becomes valid during the next count period.

Figure 6-39. Block Diagram of Operation as External Event Counter



Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
pq = 02, 04, 07

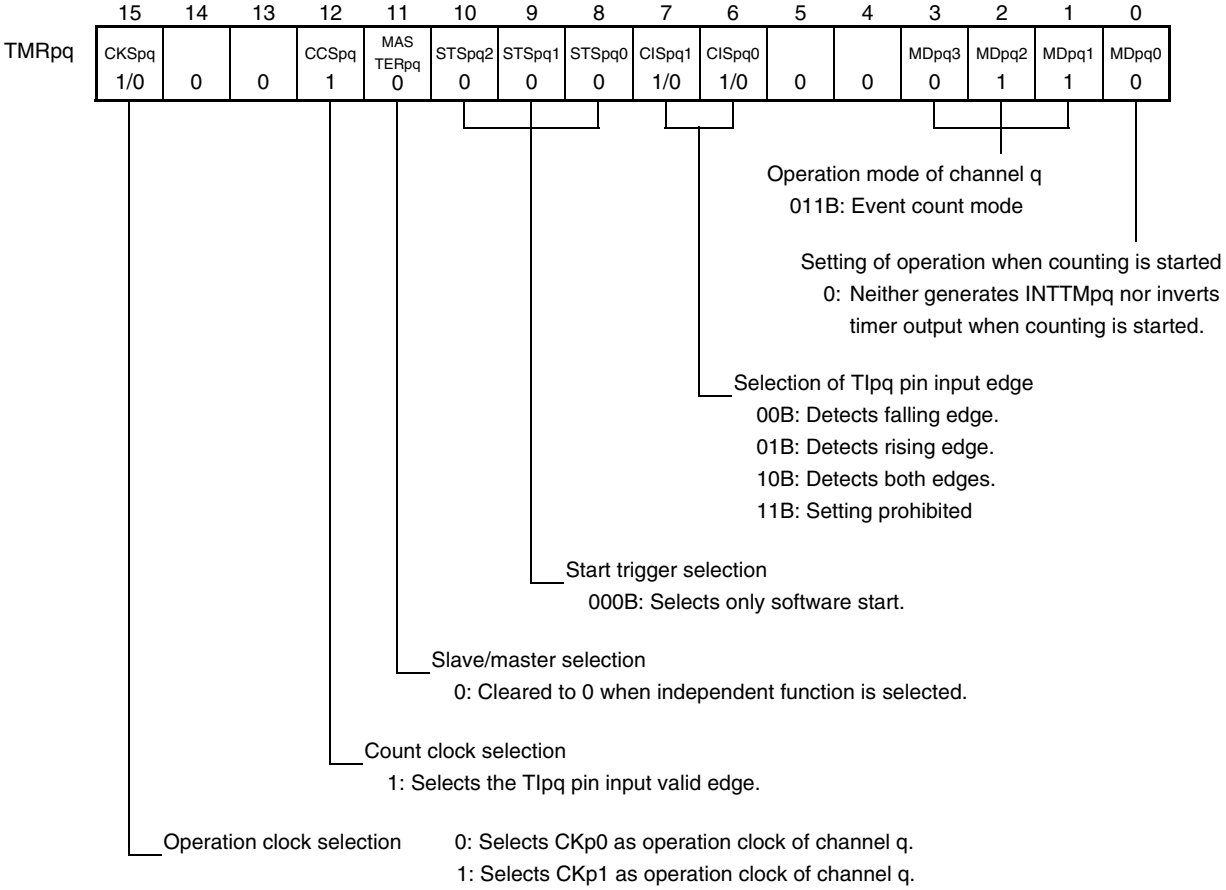
Figure 6-40. Example of Basic Timing of Operation as External Event Counter



Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
pq = 02, 04, 07

Figure 6-41. Example of Set Contents of Registers in External Event Counter Mode

(a) Timer mode register pq (TMRpq)



(b) Timer output register p (TOp)

	Bit q	
TOp	TOpq 0	0: Outputs 0 from TOpq.

(c) Timer output enable register p (TOEp)

	Bit q	
TOEp	TOEpq 0	0: Stops the TOpq output operation by counting operation.

(d) Timer output level register p (TOLp)

	Bit q	
TOLp	TOLpq 0	0: Cleared to 0 when TOMpq = 0 (toggle mode).

(e) Timer output mode register p (TOMp)

	Bit q	
TOMp	TOMpq 0	0: Sets toggle mode.

Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
p = 0, pq = 02, 04, 07

Figure 6-42. 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 TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSp register. Determines clock frequencies of CKp0 and CKp1.	
Channel default setting	Sets the TMRpq register (determines operation mode of channel). Sets number of counts to the TDRpq register. Clears the TOEpq bit of the TOEp register to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSpq bit to 1. → The TSpq bit automatically returns to 0 because it is a trigger bit.	TEpq = 1, and count operation starts. Value of TDRpq is loaded to TCRpq and detection of the TIpq pin input edge is awaited.
During operation	Set value of the TDRpq register can be changed. The TCRpq register can always be read. The TSRpq register is not used. Set values of TMRpq, TOMp, TOLp, TOp, and TOEp registers cannot be changed.	Counter (TCRpq) counts down each time input edge of the TIpq pin has been detected. When count value reaches 0000H, the value of TDRpq is loaded to TCRpq again, and the count operation is continued. By detecting TCRpq = 0000H, the INTTMpq output is generated. After that, the above operation is repeated.
Operation stop	The TTpq bit is set to 1. → The TTpq bit automatically returns to 0 because it is a trigger bit.	TEpq = 0, and count operation stops. TCRpq holds count value and stops.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
p = 0, pq = 02, 04, 07

6.7.3 Operation as frequency divider

The timer array unit can be used as a frequency divider that divides a clock input to the TIpq pin and outputs the result from TOpq.

The divided clock frequency output from TOpq can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDRpq + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDRpq + 1)

TCRpq operates as a down counter in the interval timer mode.

After the channel start trigger bit (TSpq) is set to 1, TCRpq loads the value of TDRpq when the TIpq valid edge is detected. If MDpq0 of TMRpq = 0 at this time, INTTMpq is not output and TOpq is not toggled. If MDpq0 of TMRpq = 1, INTTMpq is output and TOpq is toggled.

After that, TCRpq counts down at the valid edge of TIpq. When TCRpq = 0000H, it toggles TOpq. At the same time, TCRpq loads the value of TDRpq again, and continues counting.

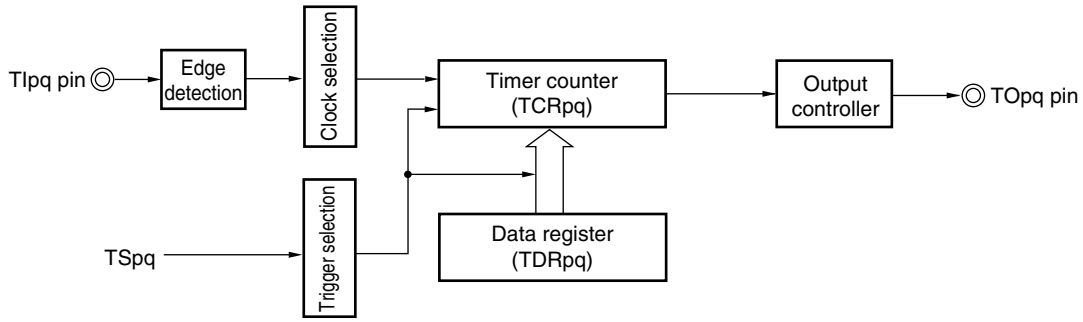
If detection of both the edges of TIpq is selected, the duty factor error of the input clock affects the divided clock period of the TOpq output.

The period of the TOpq output clock includes a sampling error of one period of the operation clock.

Clock period of TOpq output = Ideal TOpq output clock period ± Operation clock period (error)

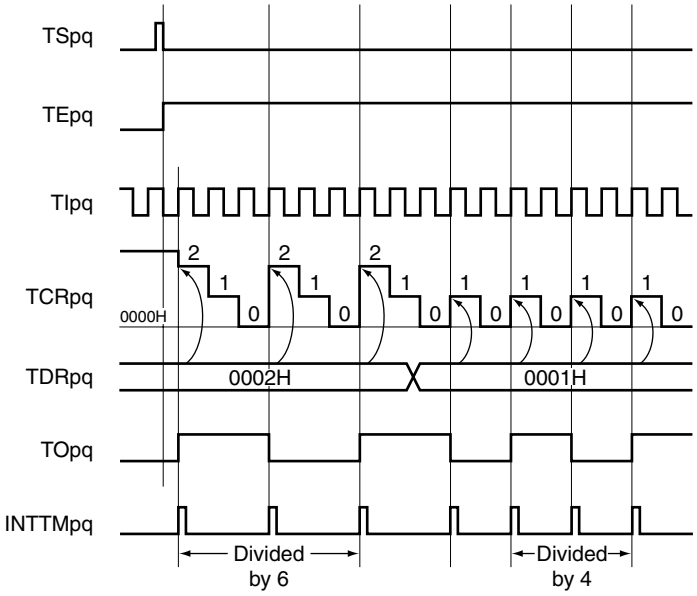
TDRpq can be rewritten at any time. The new value of TDRpq becomes valid during the next count period.

Figure 6-43. Block Diagram of Operation as Frequency Divider



Remark pq: Unit number + Channel number (only for channels provided with timer I/O pins)
pq = 02

Figure 6-44. Example of Basic Timing of Operation as Frequency Divider (MDpq0 = 1)



Remark pq: Unit number + Channel number (only for channels provided with timer I/O pins)
pq = 02

Figure 6-45. Example of Set Contents of Registers When Frequency Divider Is Used

(a) Timer mode register pq (TMRpq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRpq	CKSpq			CCSpq	MAS TERpq	STSpq2	STSpq1	STSpq0	CISpq1	CISpq0			MDpq3	MDpq2	MDpq1	MDpq0
	1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	0	0	1/0

Operation mode of channel q
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMpq nor inverts timer output when counting is started.
1: Generates INTTMpq and inverts timer output when counting is started.

Selection of TIpq pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when independent function is selected.

Count clock selection
1: Selects the TIpq pin input valid edge.

Operation clock selection
0: Selects CKp0 as operation clock of channel q.
1: Selects CKp1 as operation clock of channel q.

(b) Timer output register p (TOp)

	Bit q
TOp	TOpq
	1/0

0: Outputs 0 from TOpq.
1: Outputs 1 from TOpq.

(c) Timer output enable register p (TOEp)

	Bit q
TOEp	TOEpq
	1/0

0: Stops the TOpq output operation by counting operation.
1: Enables the TOpq output operation by counting operation.

(d) Timer output level register p (TOLp)

	Bit q
TOLp	TOLpq
	0

0: Cleared to 0 when TOMpq = 0 (toggle mode)

(e) Timer output mode register p (TOMp)

	Bit q
TOMp	TOMpq
	0

0: Sets toggle mode.

Remark pq: Unit number + Channel number (only for channels provided with timer I/O pins)
pq = 02

Figure 6-46. Operation Procedure When Frequency Divider Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSp register. Determines clock frequencies of CKp0 and CKp1.	
Channel default setting	Sets the TMRpq register (determines operation mode of channel). Sets interval (period) value to the TDRpq register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOMpq bit of the TOMp register to 0 (toggle mode). Clears the TOLpq bit to 0. Sets the TOpq bit and determines default level of the TOpq output. →	The TOpq pin goes into Hi-Z output state. The TOpq default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEpq to 1 and enables operation of TOpq. →	TOpq does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOpq pin outputs the TOpq set level.
Operation start	Sets the TOEpq to 1 (only when operation is resumed). Sets the TSpq bit to 1. → The TSpq bit automatically returns to 0 because it is a trigger bit.	TEpq = 1, and count operation starts. Value of TDRpq is loaded to TCRpq at the count clock input. INTTMpq is generated and TOpq performs toggle operation if the MDpq0 bit of the TMRpq register is 1.
During operation	Set value of the TDRpq register can be changed. The TCRpq register can always be read. The TSRpq register is not used. Set values of TOp and TOEp registers can be changed. Set values of TMRpq, TOMp, and TOLp registers cannot be changed.	Counter (TCRpq) counts down. When count value reaches 0000H, the value of TDRpq is loaded to TCRpq again, and the count operation is continued. By detecting TCRpq = 0000H, INTTMpq is generated and TOpq performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTpq bit is set to 1. → The TTpq bit automatically returns to 0 because it is a trigger bit.	TEpq = 0, and count operation stops. TCRpq holds count value and stops. The TOpq output is not initialized but holds current status.
	TOEpq is cleared to 0 and value is set to the TOp register. →	The TOpq pin outputs the TOpq set level.
TAU stop	To hold the TOpq pin output level Clears TOpq bit to 0 after the value to be held is set to the port register. →	The TOpq pin output level is held by port function.
	When holding the TOpq pin output level is not necessary Switches the port mode register to input mode. →	The TOpq pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOpq bit is cleared to 0 and the TOpq pin is set to port mode).

Operation is resumed. →

Remark pq: Unit number + Channel number (only for channels provided with timer I/O pins)
pq = 02

6.7.4 Operation as input pulse interval measurement

The count value can be captured at the TIpq valid edge and the interval of the pulse input to TIpq can be measured.

The pulse interval can be calculated by the following expression.

TIpq input pulse interval = Period of count clock × ((10000H × TSRpq: OVF) + (Capture value of TDRpq + 1))

Caution The TIpq pin input is sampled using the operating clock selected with the CKSpq bit of the TMRpq register, so an error equal to the number of operating clocks occurs.

TCRpq operates as an up counter in the capture mode.

When the channel start trigger (TSpq) is set to 1, TCRpq counts up from 0000H in synchronization with the count clock.

When the TIpq pin input valid edge is detected, the count value is transferred (captured) to TDRpq and, at the same time, the counter (TCRpq) is cleared to 0000H, and the INTTMpq is output. If the counter overflows at this time, the OVFpq bit of the TSRpq register is set to 1. If the counter does not overflow, the OVFpq bit is cleared. After that, the above operation is repeated.

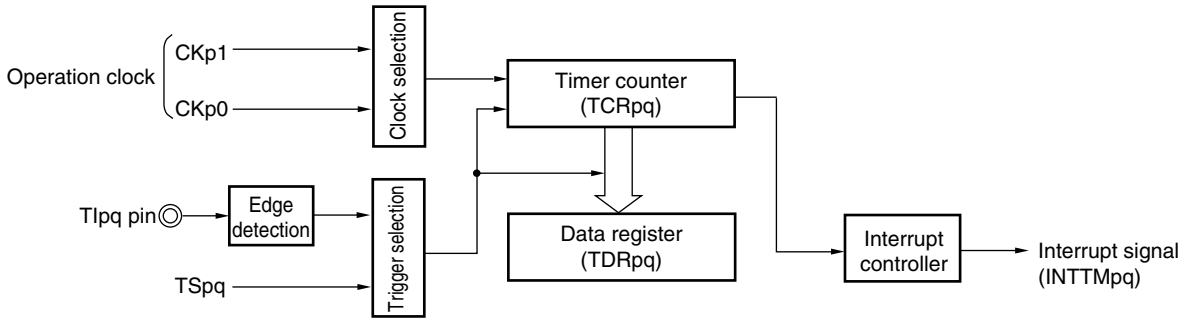
As soon as the count value has been captured to the TDRpq register, the OVFpq bit of the TSRpq register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVFpq bit of the TSRpq register is set to 1. However, the OVFpq bit is configured as a cumulative flag, the correct interval value cannot be measured if an overflow occurs more than once.

Set STSpq2 to STSpq0 of the TMRpq register to 001B to use the valid edges of TIpq as a start trigger and a capture trigger.

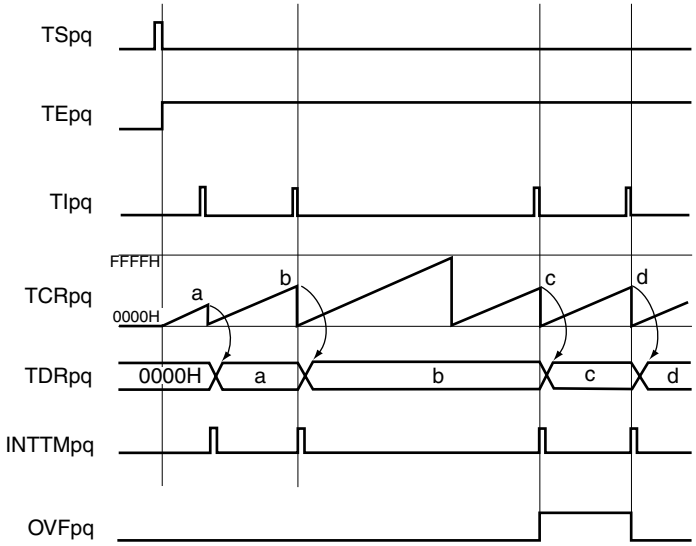
When TEpq = 1, instead of the TIpq pin input, a software operation (TSpq = 1) can be used as a capture trigger.

Figure 6-47. Block Diagram of Operation as Input Pulse Interval Measurement



Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
p = 0, pq = 02, 04, 07

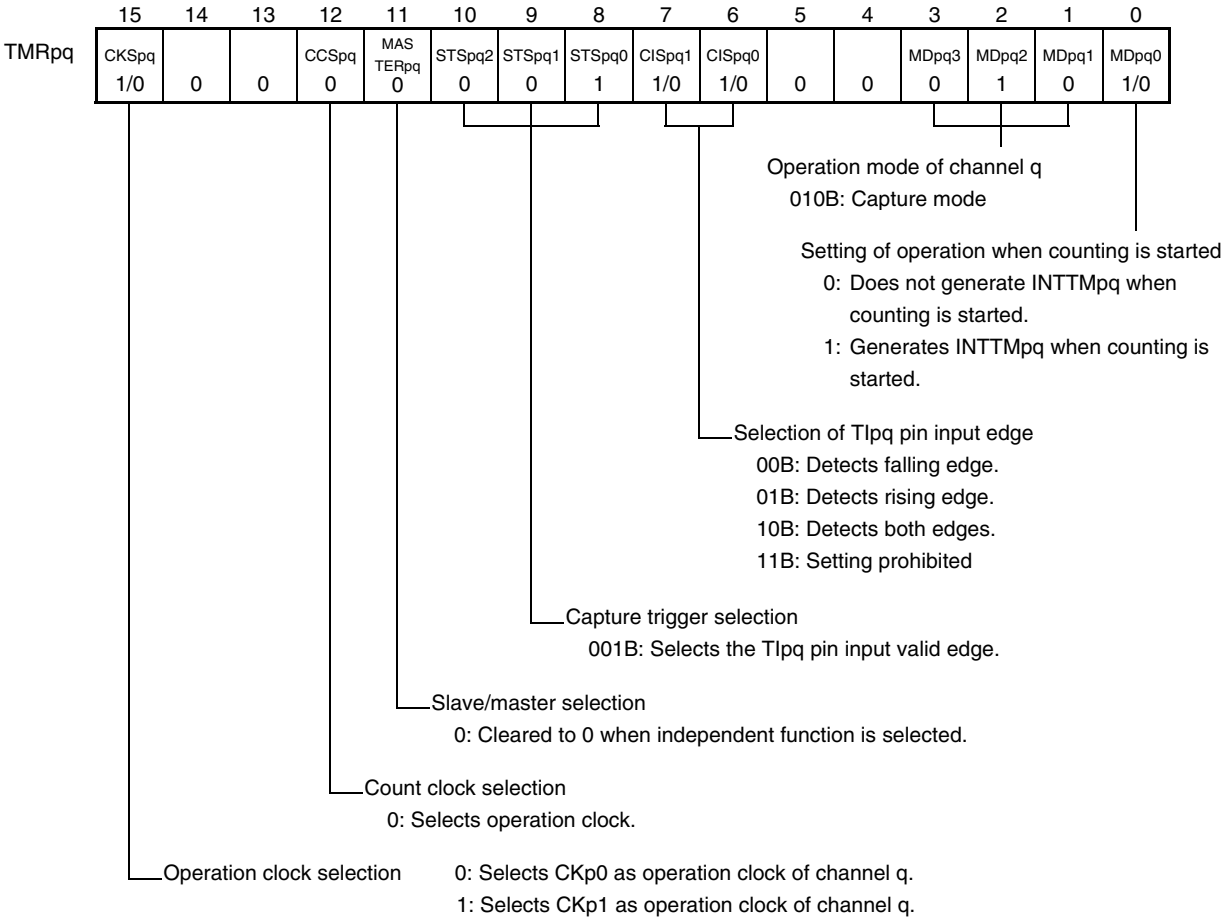
Figure 6-48. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDpq0 = 0)



Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
pq = 02, 04, 07

Figure 6-49. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register pq (TMRpq)



(b) Timer output register p (TOp)

	Bit q	
TOp	TOpq 0	0: Outputs 0 from TOpq.

(c) Timer output enable register p (TOEp)

	Bit q	
TOEp	TOEpq 0	0: Stops TOpq output operation by counting operation.

(d) Timer output level register p (TOLp)

	Bit q	
TOLp	TOLpq 0	0: Cleared to 0 when TOMpq = 0 (toggle mode).

(e) Timer output mode register p (TOMp)

	Bit q	
TOMp	TOMpq 0	0: Sets toggle mode.

Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
p = 0, pq = 02, 04, 07

Figure 6-50. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSp register. Determines clock frequencies of CKp0 and CKp1.	
Channel default setting	Sets the TMRpq register (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSpq bit to 1. → The TSpq bit automatically returns to 0 because it is a trigger bit.	TEpq = 1, and count operation starts. TCRpq is cleared to 0000H at the count clock input. When the MDpq0 bit of the TMRpq register is 1, INTTMpq is generated.
During operation	Set values of only the CISpq1 and CISpq0 bits of the TMRpq register can be changed. The TDRpq register can always be read. The TCRpq register can always be read. The TSRpq register can always be read. Set values of TOMp, TOLp, TOp, and TOEp registers cannot be changed.	Counter (TCRpq) counts up from 0000H. When the TIpq pin input valid edge is detected, the count value is transferred (captured) to TDRpq. At the same time, TCRpq is cleared to 0000H, and the INTTMpq signal is generated. If an overflow occurs at this time, the OVFpq bit of the TSRpq register is set; if an overflow does not occur, the OVFpq bit is cleared. After that, the above operation is repeated.
Operation stop	The TTpq bit is set to 1. → The TTpq bit automatically returns to 0 because it is a trigger bit.	TEpq = 0, and count operation stops. TCRpq holds count value and stops. The OVFpq bit of the TSRpq register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
p = 0, pq = 02, 04, 07

6.7.5 Operation as input signal high-/low-level width measurement

By starting counting at one edge of TIpq and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TIpq can be measured. The signal width of TIpq can be calculated by the following expression.

Signal width of TIpq input = Period of count clock × ((10000H × TSRpq: OVF) + (Capture value of TDRpq + 1))

Caution The TIpq pin input is sampled using the operating clock selected with the CKSpq bit of the TMRpq register, so an error equal to the number of operating clocks occurs.

TCRpq operates as an up counter in the capture & one-count mode.

When the channel start trigger (TSpq) is set to 1, TEpq is set to 1 and the TIpq pin start edge detection wait status is set.

When the TIpq start valid edge (rising edge of TIpq when the high-level width is to be measured) is detected, the counter counts up in synchronization with the count clock. When the valid capture edge (falling edge of TIpq when the high-level width is to be measured) is detected later, the count value is transferred to TDRpq and, at the same time, INTTMpq is output. If the counter overflows at this time, the OVFpq bit of the TSRpq register is set to 1. If the counter does not overflow, the OVFpq bit is cleared. TCRpq stops at the value “value transferred to TDRpq + 1”, and the TIpq 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 TDRpq register, the OVFpq bit of the TSRpq 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 OVFpq bit of the TSRpq register is set to 1. However, the OVFpq bit is configured as an integral flag, and the correct interval value cannot be measured if an overflow occurs more than once.

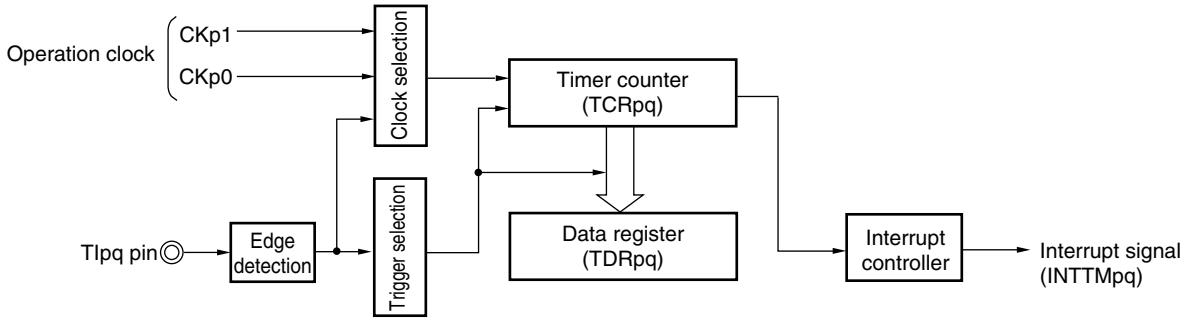
Whether the high-level width or low-level width of the TIpq pin is to be measured can be selected by using the CISpq1 and CISpq0 bits of the TMRpq register.

Because this function is used to measure the signal width of the TIpq pin input, TSpq cannot be set to 1 while TEpq is 1.

CISpq1, CISpq0 of TMRpq = 10B: Low-level width is measured.
CISpq1, CISpq0 of TMRpq = 11B: High-level width is measured.

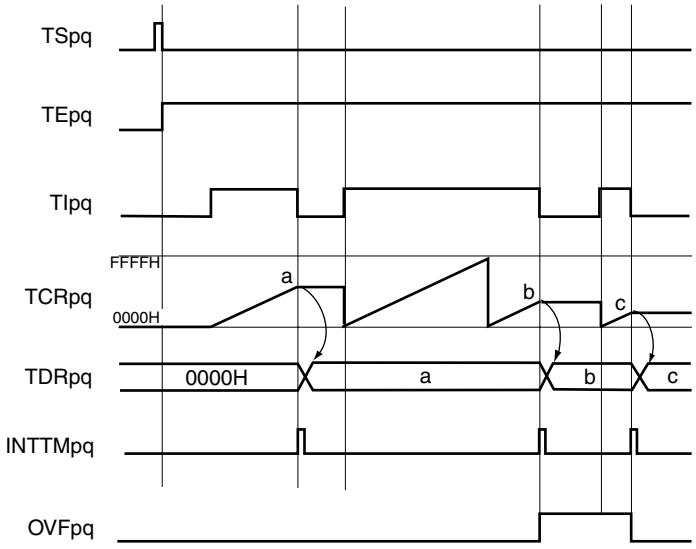
Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
pq = 02, 04, 07

Figure 6-51. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
p = 0, pq = 02, 04, 07

Figure 6-52. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
pq = 02, 04, 07

Figure 6-53. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register pq (TMRpq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRpq	CKSpq			CCSpq	MAS TERpq	STSpq2	STSpq1	STSpq0	CISpq1	CISpq0			MDpq3	MDpq2	MDpq1	MDpq0
	1/0	0	0	0	0	0	1	0	1	1/0	0	0	1	1	0	0

Operation mode of channel q
110B: Capture & one-count

Setting of operation when counting is started
0: Does not generate INTTMpq when counting is started.

Selection of TIpq pin input edge
10B: Both edges (to measure low-level width)
11B: Both edges (to measure high-level width)

Start trigger selection
010B: Selects the TIpq pin input valid edge.

Slave/master selection
0: Cleared to 0 when independent function is selected.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CKp0 as operation clock of channel q.
1: Selects CKp1 as operation clock of channel q.

(b) Timer output register p (TOp)

	Bit q
TOp	TOpq
	0

0: Outputs 0 from TOpq.

(c) Timer output enable register p (TOEp)

	Bit q
TOEp	TOEpq
	0

0: Stops the TOpq output operation by counting operation.

(d) Timer output level register p (TOLp)

	Bit q
TOLp	TOLpq
	0

0: Cleared to 0 when TOMpq = 0 (toggle mode).

(e) Timer output mode register p (TOMp)

	Bit q
TOMp	TOMpq
	0

0: Sets toggle mode.

Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
p = 0, pq = 02, 04, 07

Figure 6-54. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSp register. Determines clock frequencies of CKp0 and CKp1.	
Channel default setting	Sets the TMRpq register (determines operation mode of channel). Clears TOEpq to 0 and stops operation of TOpq.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSpq bit to 1. → The TSpq bit automatically returns to 0 because it is a trigger bit.	TEpq = 1, and the TIpq pin start edge detection wait status is set.
	Detects TIpq pin input count start valid edge. →	Clears TCRpq to 0000H and starts counting up.
During operation	Set value of the TDRpq register can be changed. The TCRpq register can always be read. The TSRpq register is not used. Set values of TMRpq, TOMp, TOLp, TOp, and TOEp registers cannot be changed.	When the TIpq pin start edge is detected, the counter (TCRpq) counts up from 0000H. If a capture edge of the TIpq pin is detected, the count value is transferred to TDRpq and INTTMpq is generated. If an overflow occurs at this time, the OVFpq bit of the TSRpq register is set; if an overflow does not occur, the OVFpq bit is cleared. TCRpq stops the count operation until the next TIpq pin start edge is detected.
Operation stop	The TTpq bit is set to 1. → TTpq bit automatically returns to 0 because it is a trigger bit.	TEpq = 0, and count operation stops. TCRpq holds count value and stops. The OVFpq bit of the TSRpq register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark pq: Unit number + Channel number (only for channels provided with timer input pins)
p = 0, pq = 02, 04, 07

6.8 Operation of Plural Channels of Timer Array Unit

6.8.1 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDRmp (slave)}/{Set value of TDRmn (master) + 1} × 100
0% output: Set value of TDRmp (slave) = 0000H
100% output: Set value of TDRmp (slave) ≥ {Set value of TDRmn (master) + 1}

Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode and counts the periods. When the channel start trigger (TSmn) is set to 1, INTTMmn is output. TCRmn counts down starting from the loaded value of TDRmn, in synchronization with the count clock. When TCRmn = 0000H, INTTMmn is output. TCRmn loads the value of TDRmn again. After that, it continues the similar operation.

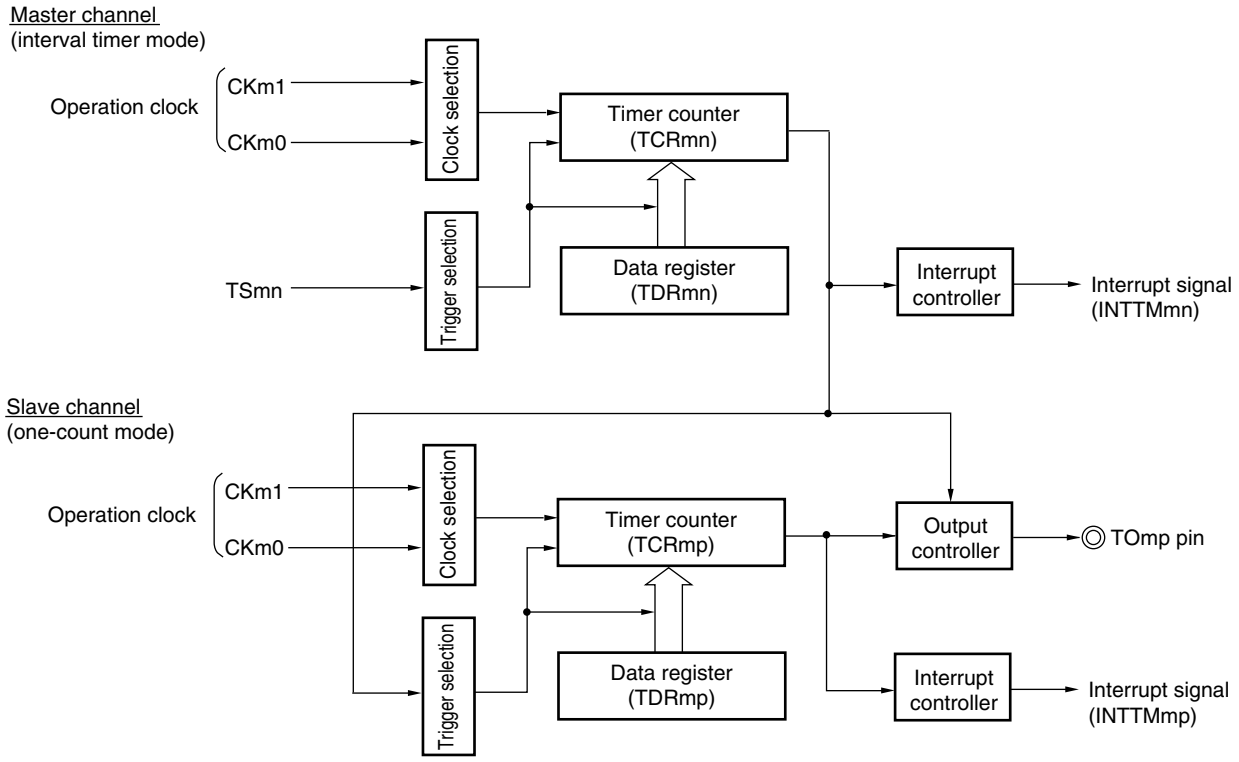
TCRmp of a slave channel operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. TCRmp of the slave channel loads the value of TDRmp, using INTTMmn of the master channel as a start trigger, and stops counting until the next start trigger (INTTMmn of the master channel) is input.

The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Caution To rewrite both TDRmn of the master channel and TDRmp of the slave channel, a write access is necessary two times. The timing at which the values of TDRmn and TDRmp are loaded to TCRmn and TCRmp 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 TDRmn of the master and TDRmp of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.

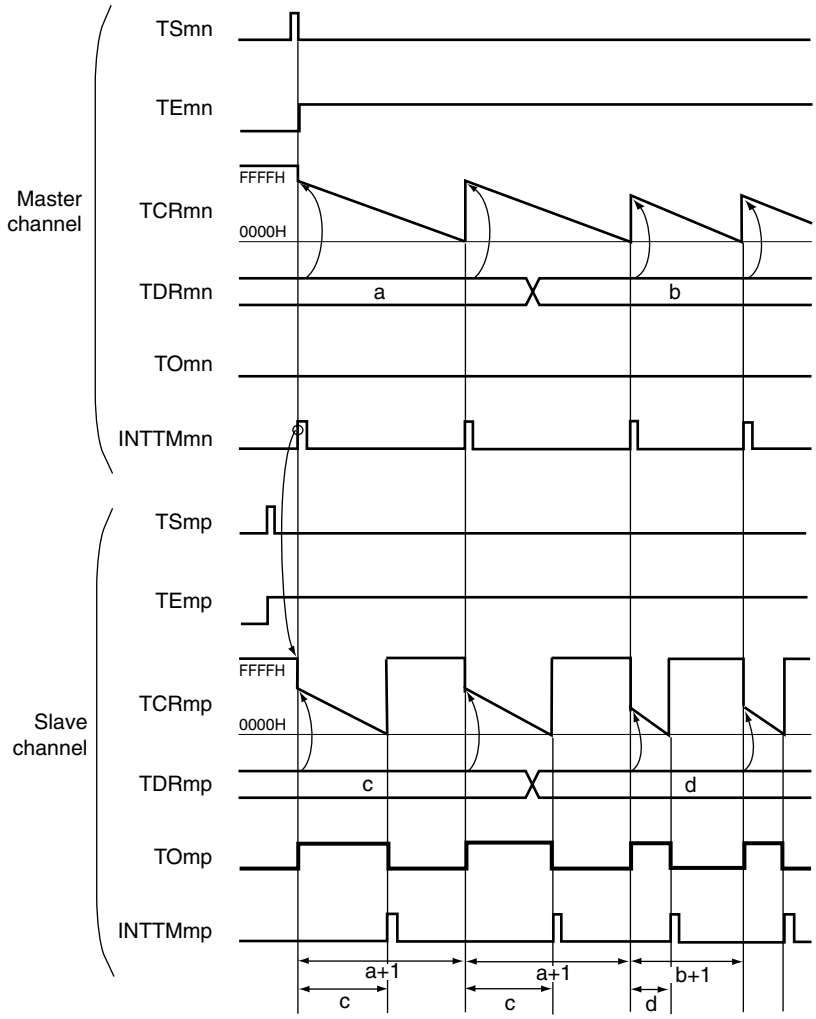
Remark m = 0, n = 4, 6, p = 5, 7, TO05, and TO07 pins

Figure 6-55. Block Diagram of Operation as PWM Function



Remark m = 0, n = 4, 6, p = 5, 7, TO05, and TO07 pins

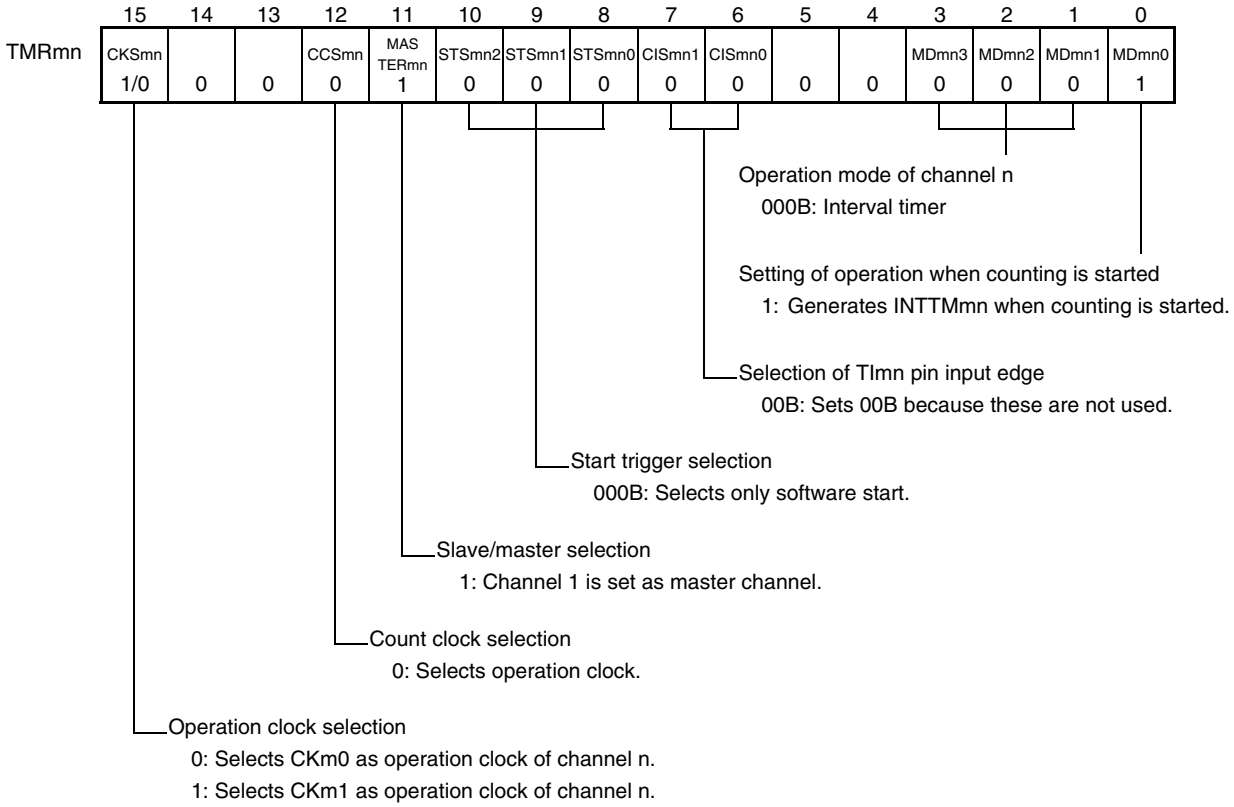
Figure 6-56. Example of Basic Timing of Operation as PWM Function



Remark m = 0, n = 4, 6, p = 5, 7, TO05, and TO07 pins

Figure 6-57. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (toggle mode).

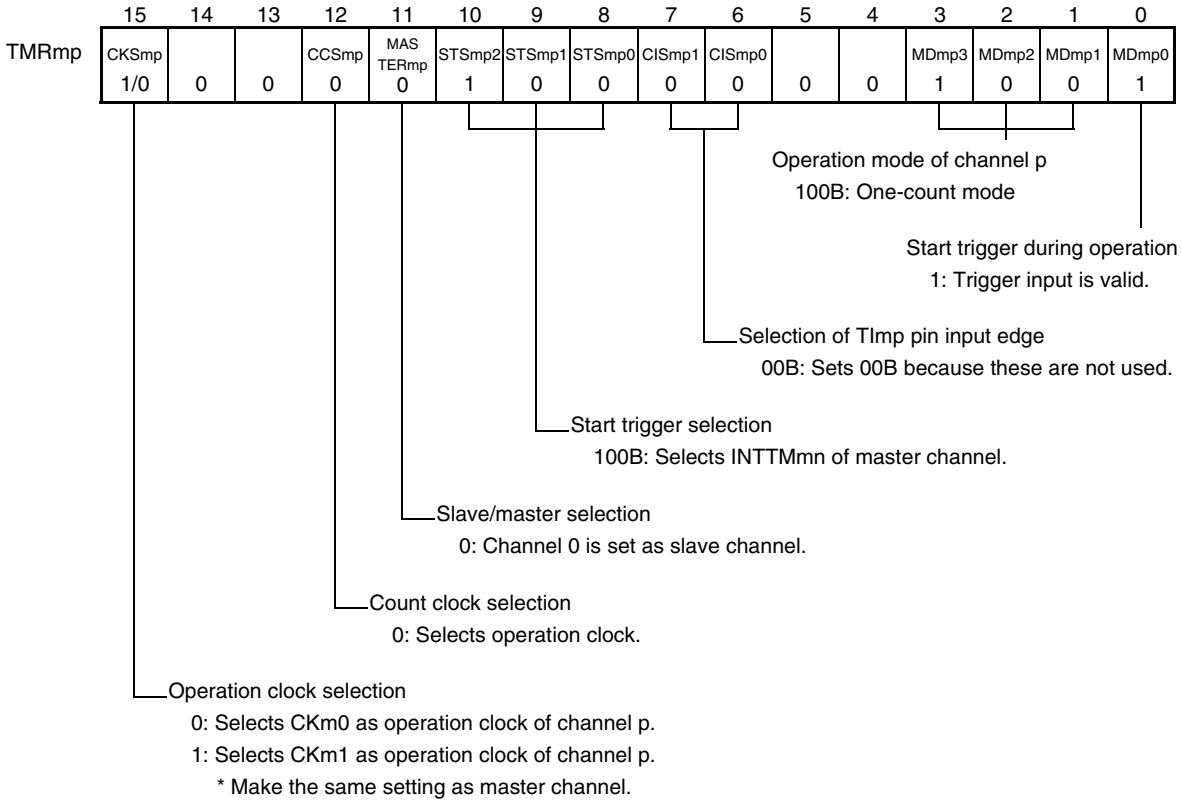
(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets toggle mode.

Remark m = 0, n = 4, 6, p = 5, 7, TO05, and TO07 pins

Figure 6-58. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register mp (TMRmp)



(b) Timer output register m (TOm)

	Bit p	
TOm	TOmp 1/0	0: Outputs 0 from TOmp. 1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p	
TOEm	TOEmp 1/0	0: Stops the TOmp output operation by counting operation. 1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p	
TOLm	TOLmp 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p	
TOMm	TOMmp 1	1: Sets the combination operation mode.

Remark m = 0, n = 4, 6, p = 5, 7, TO05, and TO07 pins

Figure 6-59. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the TMRmn and TMRmp registers of two channels to be used (determines operation mode of channels). An interval (period) value is set to the TDRmn register 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 the TOMm register is set to 1 (combination operation mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmn pin goes into Hi-Z output state. The TOmn default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEmp to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.
Operation start	Sets TOEmp (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of the TSm register are set to 1 at the same time. → The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRmn and TMRmp registers and 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. Set values of the TOm and TOEm registers cannot be changed.	The counter of the master channel loads the TDRmn value to 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 TCRmn, and the counter starts counting down again. At the slave channel, the value of TDRmp is loaded to TCRmp, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. TCRmn and TCRmp hold count value and stops. The TOmp output is not initialized but holds current status.
	TOEmp of slave channel is cleared to 0 and value is set to the TOmp register. →	The TOmp pin outputs the TOmp set level.

Operation is resumed.

Figure 6-59. Operation Procedure When PWM Function Is Used (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOmp pin output levels Clears TOmp bit to 0 after the value to be held is set to the port register. →	The TOmp pin output levels is held by port function.
	When holding the TOmp pin output levels is not necessary Switches the port mode register to input mode. →	The TOmp pin output levels go are into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m = 0, n = 4, 6, p = 5, 7, TO05, and TO07 pins

6.8.2 Operation as one-shot pulse output function

A one-shot pulse with any delay pulse width can be generated by using two channels in combination and TImn pin input or software manipulation (TSmn = 1).

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDRmn (master) + 2} × Count clock period
Pulse width = {Set value of TDRmp (slave)} × Count clock period

The Master channel operates in the one-count mode and counts the delays. TCRmn of the master channel starts operating upon start trigger detection and TCRmn loads the value of TDRmn. TCRmn counts down from the value of TDRmn 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.

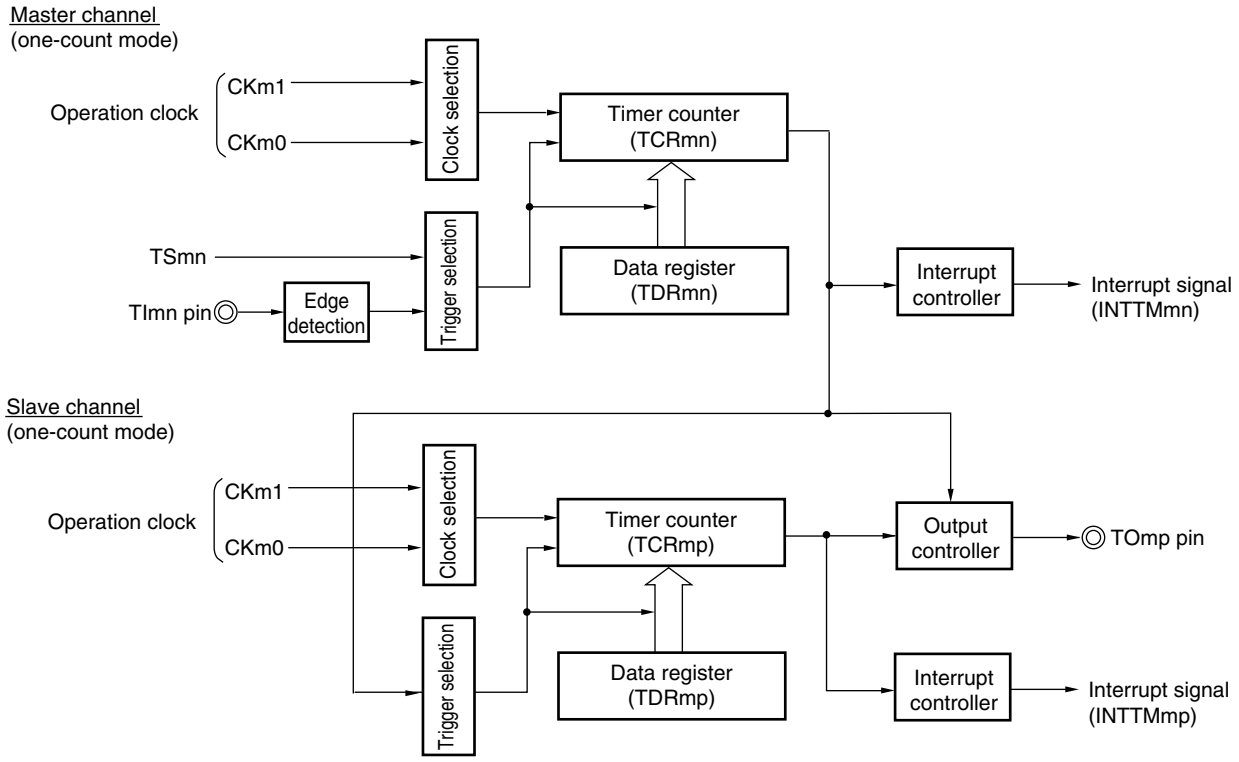
The slave channel operates in the one-count mode and counts the pulse width. TCRmp of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the TDRmp value. TCRmp counts down from the value of TDRmp it has loaded, in synchronization with the count value. When TCRmp = 0000H, it outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

Caution The timing of loading of TDRmn of the master channel is different from that of TDRmp of the slave channel. If TDRmn and TDRmp are rewritten during operation, therefore, an illegal waveform is output. Be sure to rewrite TDRmn and TDRmp after INTTMmn of the channel to be rewritten is generated.

Remark m = 0, n = 4, 6, p = 5, 7, TO05, and TO07 pins

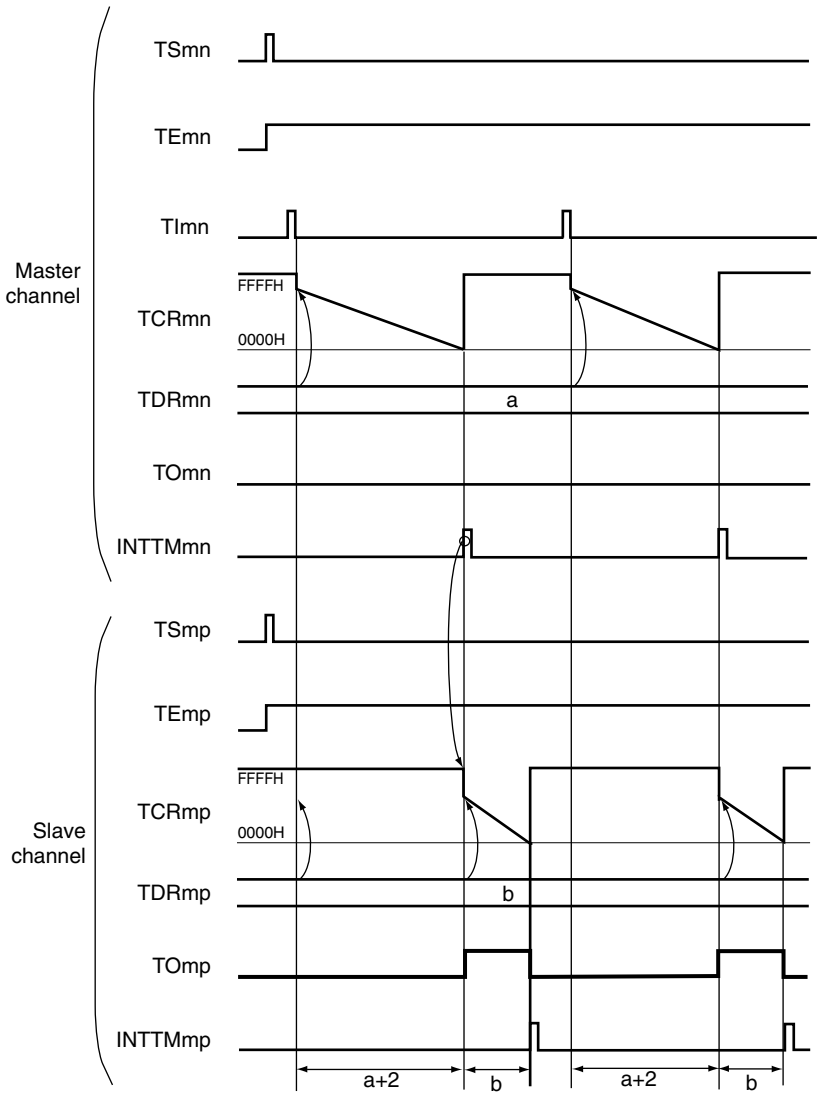
Figure 6-60. Block Diagram of Operation as One-Shot Pulse Output Function



Remark m = 0, n = 4, 6, p = 5, 7, TO05, and TO07 pins

Figure 6-61. Example of Basic Timing of Operation as One-Shot Pulse Output Function (Start Trigger TImn Input Valid Edge)

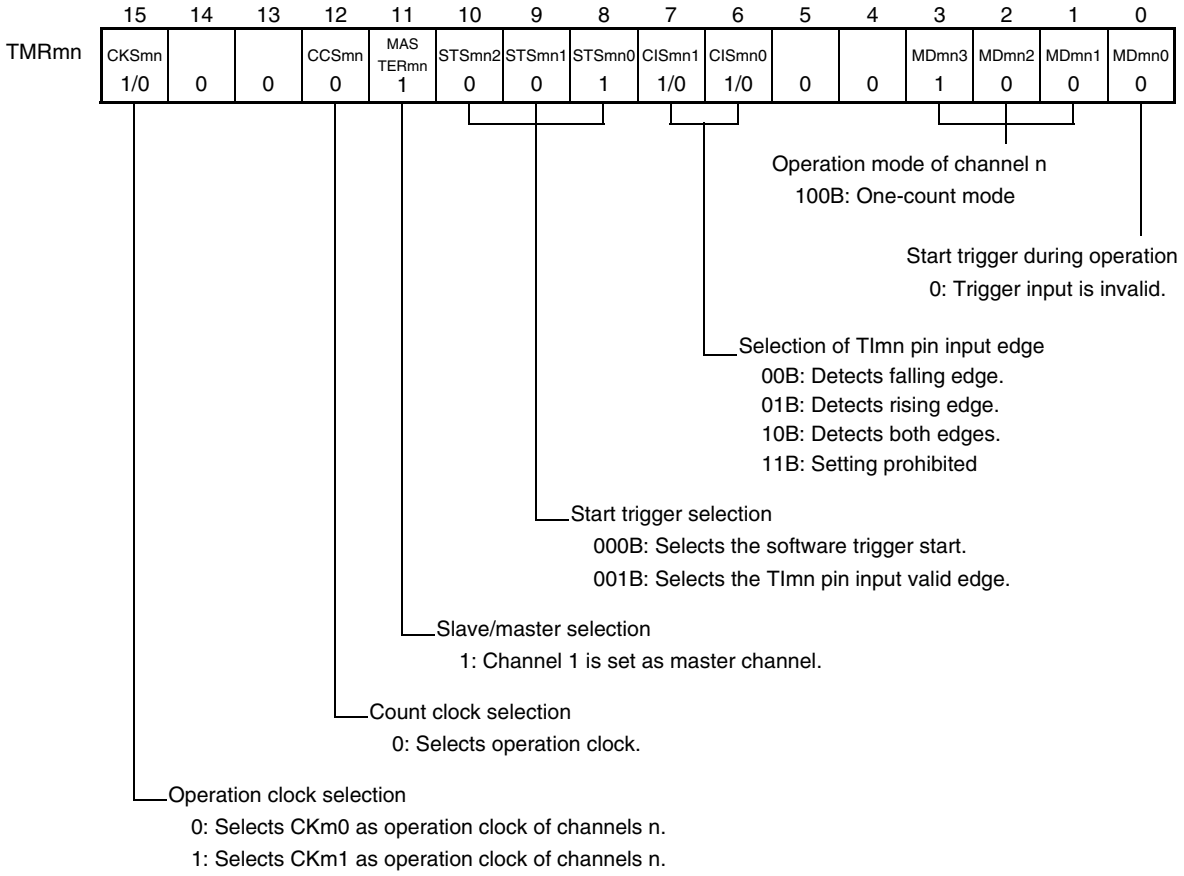
<R>



Remark m = 0, n = 4, 6, p = 5, 7, TO05, and TO07 pins

Figure 6-62. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (toggle mode).

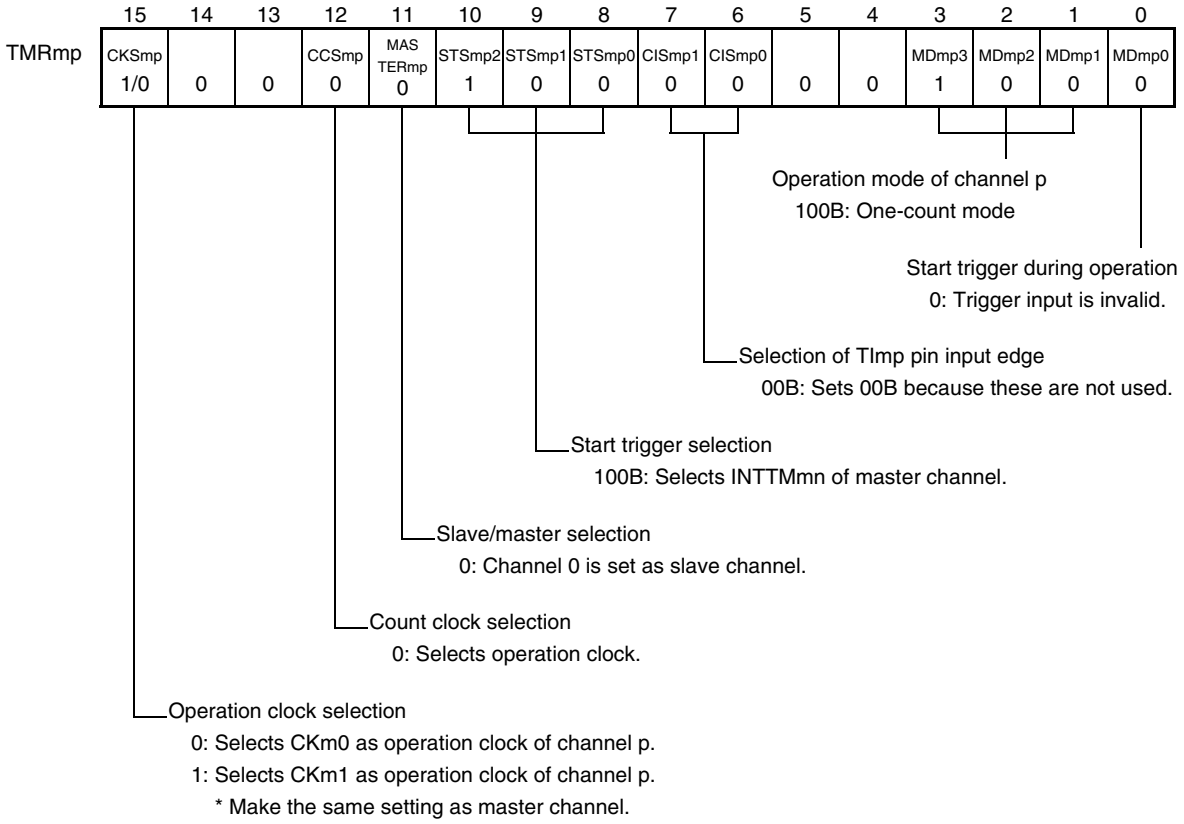
(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets toggle mode.

Remark m = 0, n = 4, 6, p = 5, 7, TO05, and TO07 pins

Figure 6-63. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register mp (TMRmp)



(b) Timer output register m (TOm)

	Bit p	
TOm	TOmp 1/0	0: Outputs 0 from TOmp. 1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p	
TOEm	TOEmp 1/0	0: Stops the TOmp output operation by counting operation. 1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p	
TOLm	TOLmp 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p	
TOMm	TOMmp 1	1: Sets the combination operation mode.

Remark m = 0, n = 4, 6, p = 5, 7, TO05, and TO07 pins

Figure 6-64. 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 TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the TMRmn and TMRmp registers of two channels to be used (determines operation mode of channels). An output delay is set to the TDRmn register 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 the TOMm register is set to 1 (combination operation mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmn pin goes into Hi-Z output state. The TOmn default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEmp to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.
Operation start	Sets TOEmp (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of the TSm register are set to 1 at the same time. → The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	TEmn and TEmp are set to 1 and the master channel enters the TImn input edge detection wait status. Counter stops operating.
	Detects the start trigger of master channel. → (The valid edge of the TImn pin input is detected or the TSmn bit is set to 1.)	Master channel starts counting.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. Set values of the TMRmp, TDRmn, and TDRmp registers and TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers can be changed.	Master channel loads the value of TDRmn to TCRmn when the start trigger is detected, 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. The slave channel, triggered by INTTMmn of the master channel, loads the value of TDRmp to TCRmp, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. TCRmn and TCRmp hold count value and stops. The TOmp output is not initialized but holds current status.
	TOEmp of slave channel is cleared to 0 and value is set to the TOm register. →	The TOmp pin outputs the TOmn set level.

Operation is resumed.

Figure 6-64. Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOmp pin output levels Clears TOmp bit to 0 after the value to be held is set to the port register. →	The TOmp pin output levels is held by port function.
	When holding the TOmp pin output levels is not necessary Switches the port mode register to input mode.	The TOmp pin output levels go are into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m = 0, n = 4, 6, p = 5, 7, TO05, and TO07 pins

CHAPTER 7 REAL-TIME COUNTER

Caution The real-time counter is only initialized by a reset triggered by the LRESET pin. Other than $\overline{\text{RESET}}$ pin, the counter is not initialized by internal resets triggered by the watchdog timer, power-on-clear (POC) circuit, low-voltage detector (LVI), or unauthorized command.

7.1 Functions of Real-Time Counter

The real-time counter has the following features.

- Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
- Constant-period interrupt function (period: 1 month to 0.5 seconds)
- Alarm interrupt function (alarm: week, hour, minute)
- Interval interrupt function
- Pin output function of 1 Hz or 32.768 kHz

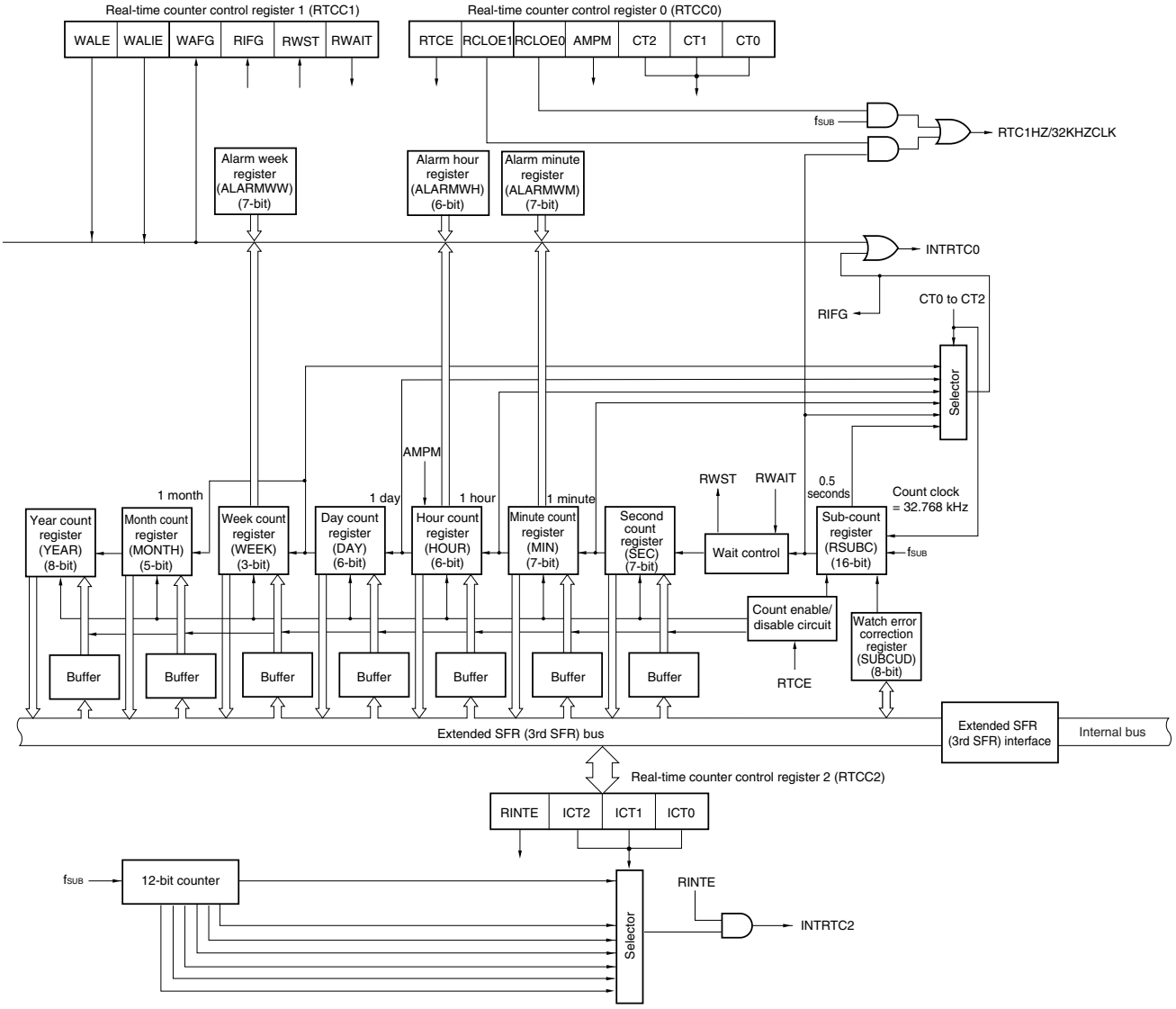
7.2 Configuration of Real-Time Counter

The real-time counter includes the following hardware.

Table 7-1. Configuration of Real-Time Counter

Item	Configuration
Control registers	Real-time counter control register 0 (RTCC0) Real-time counter control register 1 (RTCC1) Real-time counter control register 2 (RTCC2) Real-time counter mode register (RTCMD) Pull-down status control register (PUTCTL) Sub-count register (RSUBC) Second count register (SEC) Minute count register (MIN) Hour count register (HOUR) Day count register (DAY) Week count register (WEEK) Month count register (MONTH) Year count register (YEAR) Watch error correction register (SUBCUD) Alarm minute register (ALARMWM) Alarm hour register (ALARMWH) Alarm week register (ALARMWW)

Figure 7-1. Block Diagram of Real-Time Counter



7.3 Registers Controlling Real-Time Counter

The real-time counter is controlled by the following registers.

These registers are all allocated to the extended SFR (3rd SFR) space.

For details about how to access the extended SFR (3rd SFR) space, see **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE**.

- Real-time counter control register 0 (RTCC0)
- Real-time counter control register 1 (RTCC1)
- Real-time counter control register 2 (RTCC2)
- Real-time counter mode register (RTCMD)
- Pull-down status control register (PUTCTL)
- Sub-count register (RSUBC)
- 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)

(1) Real-time counter control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time counter operation, control the RTC1HZ and 32KHZCLK pins, and set a 12- or 24-hour system and the constant-period interrupt function.

RTCC0 is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 7-2. Format of Real-Time Counter Control Register 0 (RTCC0)

Address: 96H After reset: 00H R/W

<R>

Symbol	7	6	5	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1	RCLOE0	AMPM	CT2	CT1	CT0

RTCE	Real-time counter operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1[Note]	RTC1HZ pin output control
0	Disables output of RTC1HZ pin (1 Hz).
1	Enables output of RTC1HZ pin (1 Hz).

RCLOE0[Note]	32KHZCLK pin output control
0	Disables output of 32KHZCLK pin (32.768 kHz).
1	Enables output of 32KHZCLK pin (32.768 kHz).

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
Rewrite the AMPM value after setting RWAIT (bit 0 of RTCC1) to 1. If the AMPM value is changed, the values of the hour count register (HOUR) change according to the specified time system. Table 7-2 shows the displayed time digits.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC0) 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 CT2 to CT0 while the counter operates (RTCE = 1), rewrite the values of CT2 to CT0 after disabling interrupt servicing INTRTC0 by using the interrupt mask flag register. Furthermore, after rewriting the values of CT2 to CT0, enable interrupt servicing after clearing the RIFG and RTC0IF flags.			

Note RCLOE0 and RCLOE1 must not be enabled at the same time.

Caution If RCLOE0 and RCLOE1 are changed when RTCE = 1, a last pulse with a narrow width may be generated on the 32.768 kHz and 1 Hz output signals.

Remark ×: don't care

(2) Real-time counter 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 is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 7-3. Format of Real-Time Counter Control Register 1 (RTCC1) (1/2)

Address: 97H After reset: 00H R/W

<R>

Symbol	7	6	5	4	3	2	1	0
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
When setting a value to the WALE bit while the counter operates (RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt servicing INTRTC0 by using the interrupt mask flag register. Furthermore, clear the WAFG and RTC0IF flags after rewriting the WALE bit. When setting each alarm register (WALIE flag of RTCC1, the ALARMWM register, the ALARMWH register, and the ALARMWW register), set match operation to be invalid (“0”) for the WALE bit.	

WALIE	Control of alarm interrupt (INTRTC0) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to “1” one 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.	

RWST	Wait status flag of real-time counter
0	Counter is operating.
1	Mode to read or write counter value
This status flag indicates whether the setting of RWAIT is valid. Before reading or writing the counter value, confirm that the value of this flag is 1.	

Figure 7-3. Format of Real-Time Counter Control Register 1 (RTCC1) (2/2)

RWAIT	Wait control of real-time counter
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value
This bit controls the operation of the counter. Be sure to write "1" to it to read or write the counter value. Because sub-count register (RSUBC) continues operation, complete reading or writing of it in 1 second, and clear this bit back to 0. When RWAIT = 1, it takes up to 1 clock (32.768 kHz) until the counter value can be read or written. If RSUBC overflows when RWAIT = 1, it counts up after RWAIT = 0. If the second count register is written, however, it does not count up because RSUBC is cleared.	

Caution The RIFG and WAFG flags may be cleared when the RTCC1 register is written by using a 1-bit manipulation instruction. Use, therefore, an 8-bit manipulation instruction in order to write to the RTCC1 register. To prevent the RIFG and WAFG flags from being cleared during writing, disable writing by setting "1" to the corresponding bit. When the value may be rewritten because the RIFG and WAFG flags are not being used, the RTCC1 register may be written by using a 1-bit manipulation instruction.

Remark Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC0). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC0 occurrence.

(3) Real-time counter control register 2 (RTCC2)

The RTCC2 register is an 8-bit register that is used to control the interval interrupt function.

RTCC2 is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 7-4. Format of Real-Time Counter Control Register 2 (RTCC2)

Address: 92H After reset: 00H R/W

<R>

Symbol	7	6	5	4	3	2	1	0
RTCC2	RINTE	0	0	0	0	ICT2	ICT1	ICT0

RINTE	ICT2	ICT1	ICT0	Interval interrupt (INTRTC2) selection
0	×	×	×	Interval interrupt is not generated.
1	0	0	0	$2^6/f_{SUB}$ (1.953125 ms)
1	0	0	1	$2^7/f_{SUB}$ (3.90625 ms)
1	0	1	0	$2^8/f_{SUB}$ (7.8125 ms)
1	0	1	1	$2^9/f_{SUB}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{SUB}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{SUB}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{SUB}$ (125 ms)

Caution Change ICT2, ICT1, and ICT0 when RINTE = 0.

(4) Real-time counter mode register (RTCMD)

The RTCMD register is an 8-bit register that is used to set an operation mode of the real-time counter.

RTCMD is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 7-5. Format of Real-time Counter Mode Register (RTCMD)

Address: CBH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RTCMD	RTCMD7	RTCMD6	RTCMD5	RTCMD4	RTCMD3	RTCMD2	RTCMD1	RTCMD0

RTCMD7 to RTCMD0	Setting of operation mode of real-time counter
01011001 (59H)	The real-time counter operates. The 24-bit ΔΣ-type A/D converter, power calculation circuit, power quality measurement circuit, and digital frequency conversion circuit stop.
Other than above	Normal operation mode

(5) Pull-down status control register (PUTCTL)

This 8-bit register is used to control the pull-down status of pins after release of reset.

PUTCTL is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 7-6. Format of Pull-down Status Control Register (PUTCTL)

Address: 88H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PUTCTL	0	0	0	0	0	0	0	PDCUT

PDCUT	Control whether the pull-down resistor is connected
0	Connects the pull-down resistor.
1	Disconnects the pull-down resistor.

Caution The extended SFR (3rd SFR) interface is used to access the extended SFR (3rd SFR) space. After release of reset, until initializing the extended SFR (3rd SFR) interface finishes, perform pull-down control to prevent the extended SFR (3rd SFR) interface status from becoming undefined. After the extended SFR (3rd SFR) interface is initialized, specify 01H for PUTCTL and cancel the pull-down status to reduce current consumption.

(6) Sub-count register (RSUBC)

The RSUBC register is a 16-bit register that counts the reference time of 1 second of the real-time counter.

Usually, it takes a value of 0000H to 7FFFH and counts 1 second with a clock of 32.768 kHz.

RSUBC is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 0000H.

Cautions 1. When a correction is made by using the watch error correction register (SUBCUD), the value may become 8000H or more.

2. This register is also cleared by reset effected by writing the second count register.

3. The value read from this register is not guaranteed if it is read during operation, because a value that is changing is read.

Figure 7-7. Format of Sub-Count Register (RSUBC)

Address: 98H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RSUBCL	SUBC7	SUBC6	SUBC5	SUBC4	SUBC3	SUBC2	SUBC1	SUBC0

Address: 99H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RSUBCH	SUBC15	SUBC14	SUBC13	SUBC12	SUBC11	SUBC10	SUBC9	SUBC8

(7) Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the sub-counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later.

Set a decimal value of 00 to 59 to this register in BCD code.

SEC is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 7-8. Format of Second Count Register (SEC)

Address: 9AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

(8) Minute count register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) 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.

MIN is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 7-9. Format of Minute Count Register (MIN)

Address: 9BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

(9) Hour count register (HOUR)

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12, 21 to 32 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) 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 counter control register 0 (RTCC0).If the value of AMPM is changed, the values of the HOUR change according to the specified time system.

HOUR is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

Figure 7-10. Format of Hour Count Register (HOUR)

Address: 9CH After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution Bit 5 (HOUR20) of HOUR indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

Table 7-2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 7-2. Displayed Time Digits

24-Hour Display (AMPM bit = 1)		12-Hour Display (AMPM bit = 0)	
Time	HOUR Register	Time	HOUR Register
0	00H	0 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	0 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.

(10) Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.
It counts up when the hour counter overflows.
This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) 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.
DAY is allocated to the extended SFR (3rd SFR) space.
Use the extended SFR (3rd SFR) interface to set up this register.
Reset signal generation clears this register to 01H.

Figure 7-11. Format of Day Count Register (DAY)

Address: 9EH After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

(11) Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.

It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 06 to this register in BCD code.

WEEK is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 7-12. Format of Week Count Register (WEEK)

Address: 9DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution 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

(12) Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) 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.

MONTH is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 01H.

Figure 7-13. Format of Month Count Register (MONTH)

Address: 9FH After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

(13) Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month counter 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 (32.768 kHz) later. Even if the month 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 99 to this register in BCD code.

YEAR is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 7-14. Format of Year Count Register (YEAR)

Address: A0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

(14) Watch error correction register (SUBCUD)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value (reference value: 7FFFH) that overflows from the sub-count register (RSUBC) to the second count register.

Rewrite the SUBCUD after disabling interrupt servicing INTRTC0 by using the interrupt mask flag register.

Furthermore, after rewriting the SUBCUD, enable interrupt servicing after clearing the interrupt request flags (RTC0IF) and the fixed-cycle interrupt status flags (RIFG).

SUBCUD is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 7-15. Format of Watch Error Correction Register (SUBCUD)

Address: A1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch error only when the second digits are at 00 (every 60 seconds).
Writing to the SUBCUD register at the following timing is prohibited. • When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H • When DEV = 1 is set: For a period of SEC = 00H	

F6	Setting of watch error correction value
0	Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.
When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected. * is 0 or 1. /F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100). Range of correction value: (when F6 = 0) 2, 4, 6, 8, … , 120, 122, 124 (when F6 = 1) –2, –4, –6, –8, … , –120, –122, –124	

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	–189.2 ppm to 189.2 ppm	–63.1 ppm to 63.1 ppm
Maximum excludes quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

Remark Set DEV to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(15) Alarm minute register (ALARMWM)

This register is used to set minutes of alarm.

ALARMWM is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 7-16. Format of Alarm Minute Register (ALARMWM)

Address: 93H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

(16) Alarm hour register (ALARMWH)

This register is used to set hours of alarm.

ALARMWH is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 7-17. Format of Alarm Hour Register (ALARMWH)

Address: 94H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution Bit 5 (WH20) of ALARMWH indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

(17) Alarm week register (ALARMWW)

This register is used to set date of alarm.

ALARMWW is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 7-18. Format of Alarm Week Register (ALARMWW)

Address: 95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

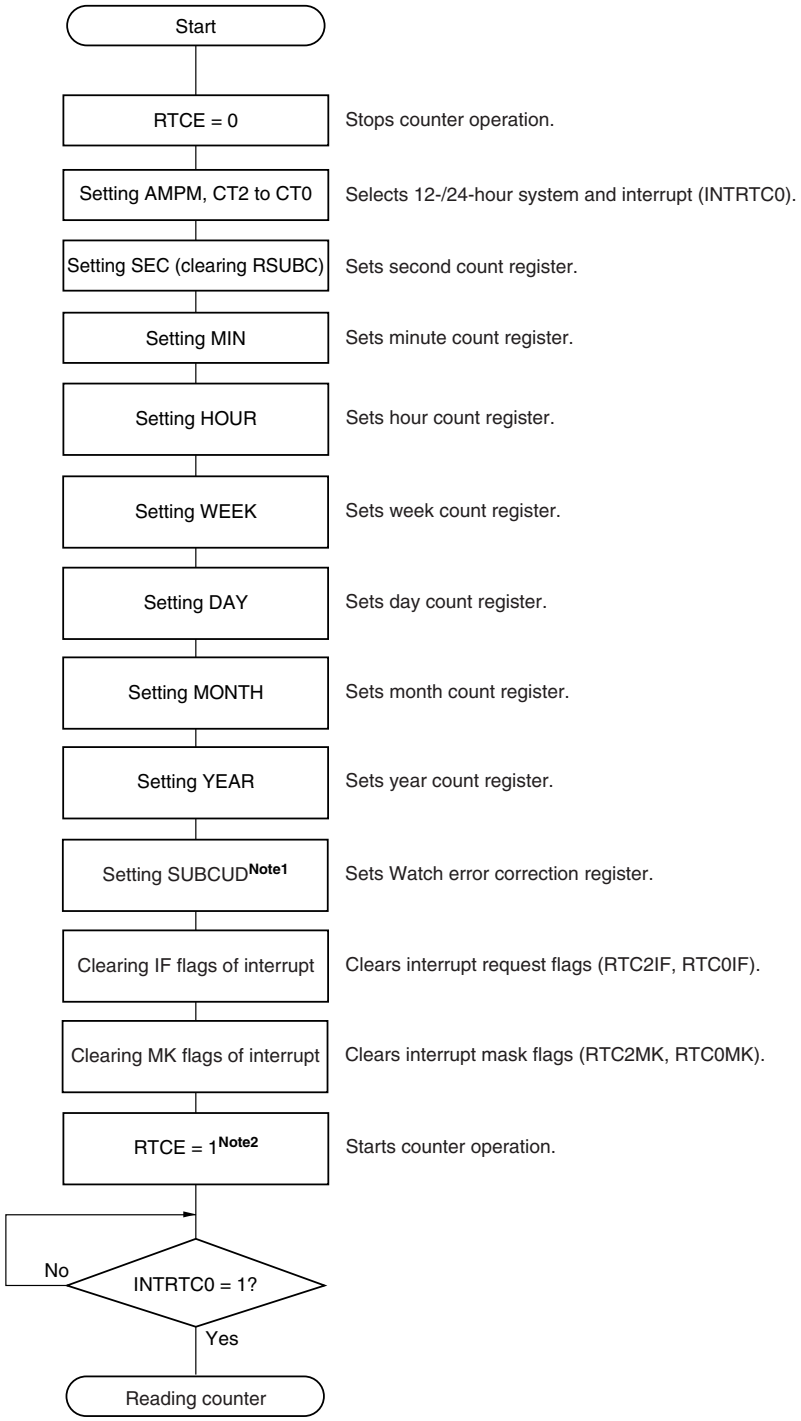
Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

7.4 Real-Time Counter Operation

7.4.1 Starting operation of real-time counter

Figure 7-19. Procedure for Starting Operation of Real-Time Counter



Notes 1. Set up SUBCUD only if the watch error must be corrected. For details about how to calculate the correction value, see **7.4.7 Example of watch error correction of real-time counter**.

2. Confirm the procedure described in **7.4.2 Shifting to STOP mode after starting operation** when shifting to STOP mode without waiting for INTRTC0 = 1 after RTCE = 1.

7.4.2 Shifting to STOP mode after starting operation

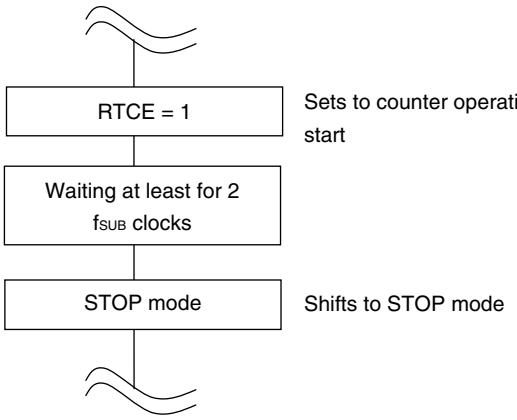
Perform one of the following processing when shifting to STOP mode immediately after setting RTCE to 1.

However, after setting RTCE to 1, this processing is not required when shifting to STOP mode after the first INTRTC0 interrupt has occurred.

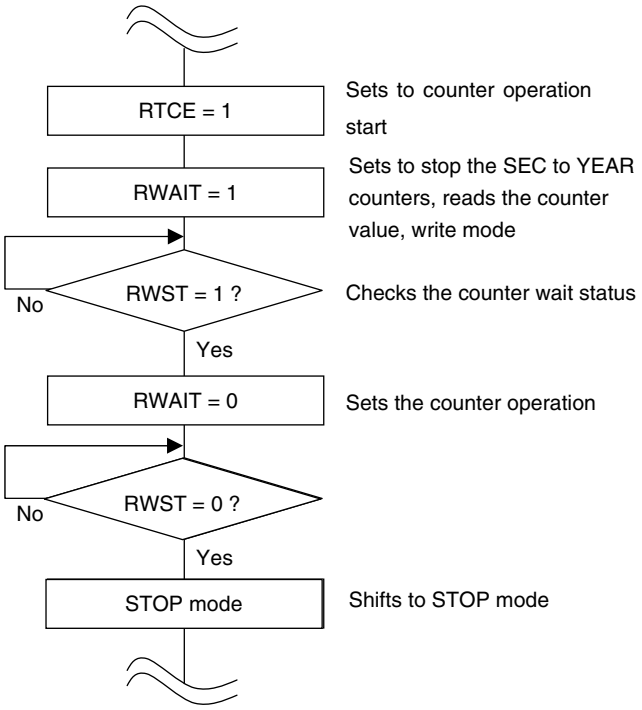
- Shifting to STOP mode when at least two input clocks (f_{SUB}) have elapsed after setting RTCE to 1 (see **Figure 7-20**, **Example 1**).
- Checking by polling RWST to become 1, after setting RTCE to 1 and then setting RWAIT to 1. Afterward, setting RWAIT to 0 and shifting to STOP mode after checking again by polling that RWST has become 0 (see **Figure 7-20**, **Example 2**).

Figure 7-20. Procedure for Shifting to STOP Mode After Setting RTCE to 1

Example 1



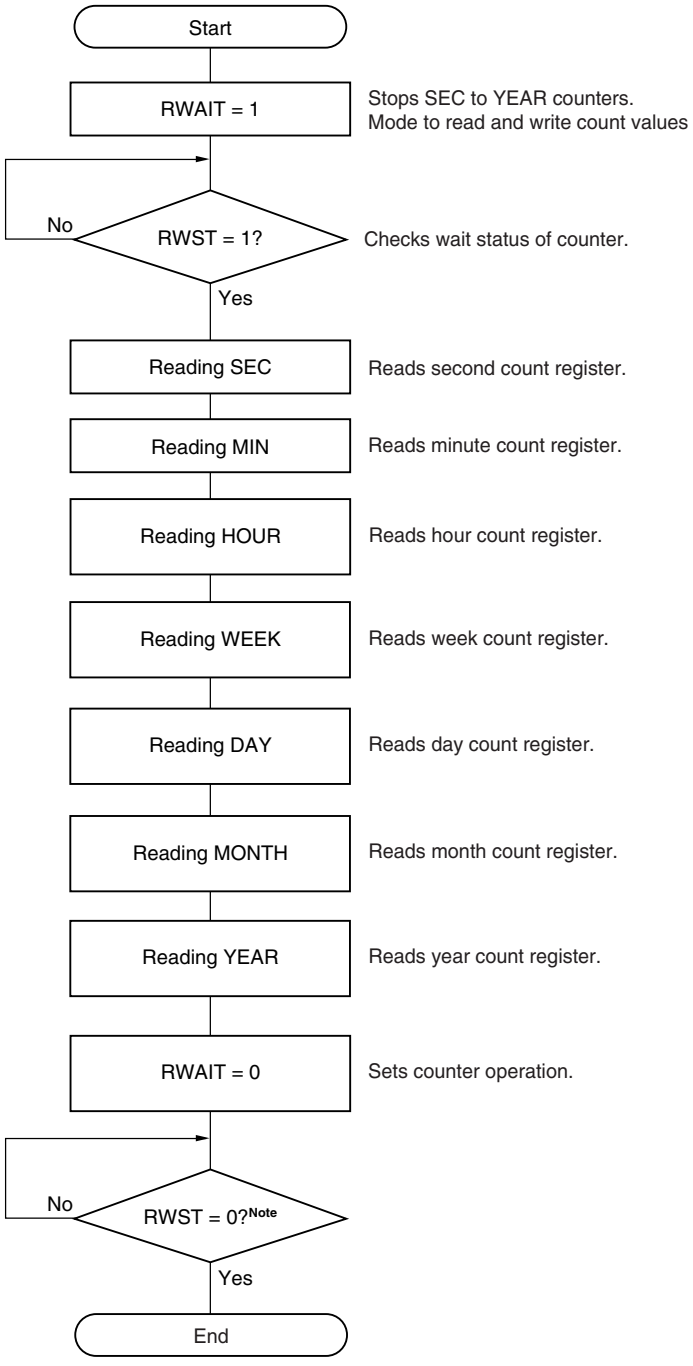
Example 2



7.4.3 Reading/writing real-time counter

Read or write the counter after setting 1 to RWAIT first.

Figure 7-21. Procedure for Reading Real-Time Counter

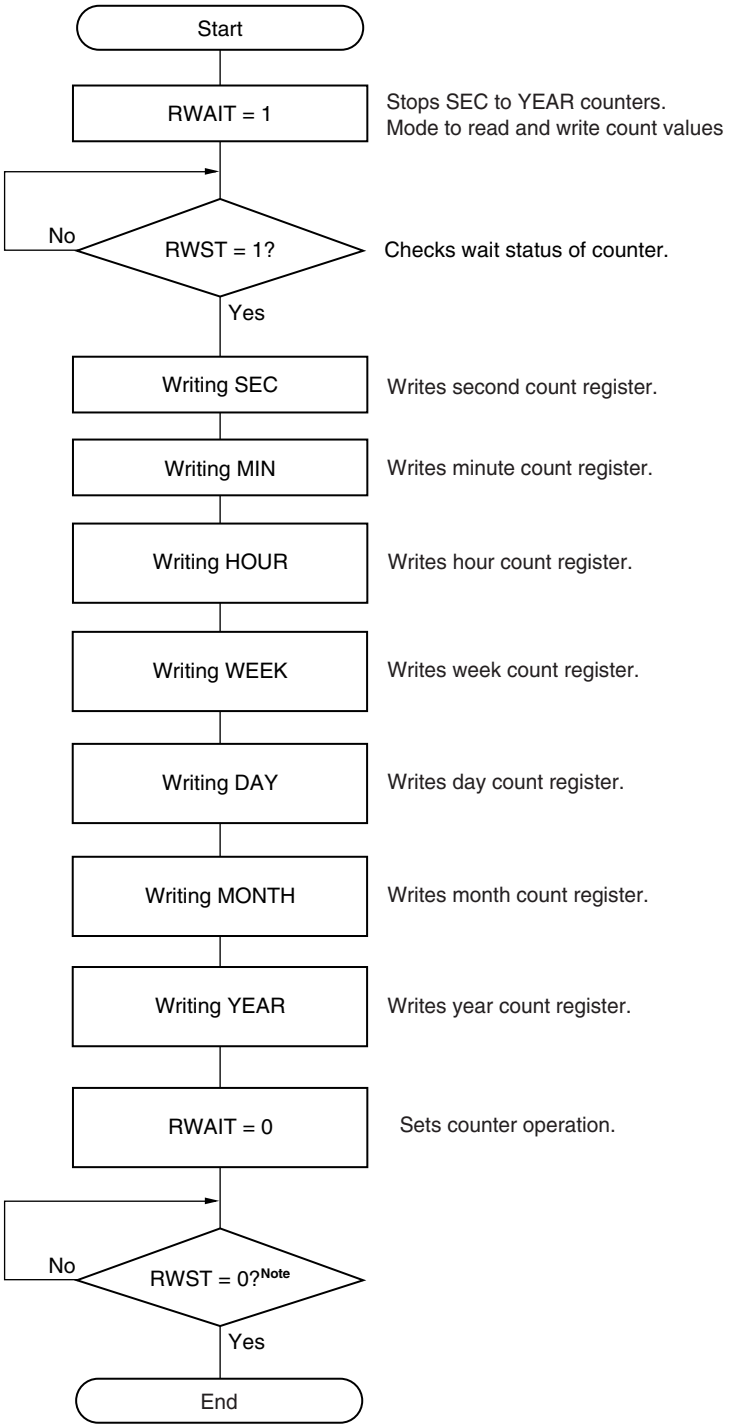


Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT 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.

Figure 7-22. Procedure for Writing Real-Time Counter



Note Be sure to confirm that RWST = 0 before setting STOP mode.

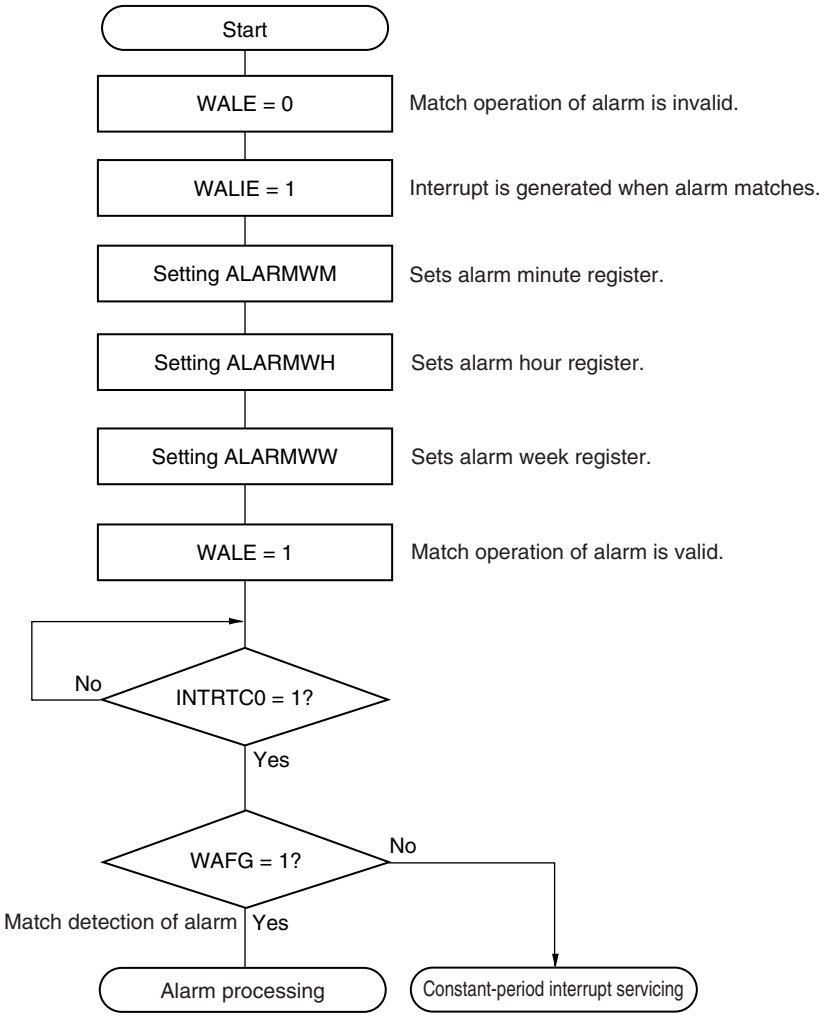
Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be written in any sequence.
All the registers do not have to be set and only some registers may be written.

7.4.4 Setting alarm of real-time counter

Set time of alarm after setting 0 to WALE first.

Figure 7-23. Alarm Setting Procedure

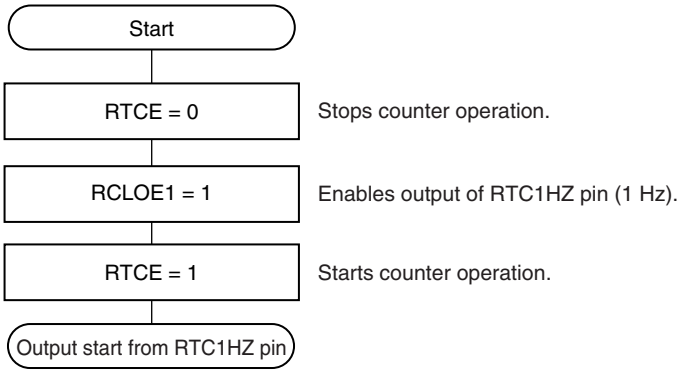


Remarks 1. ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC0). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC0 occurrence.

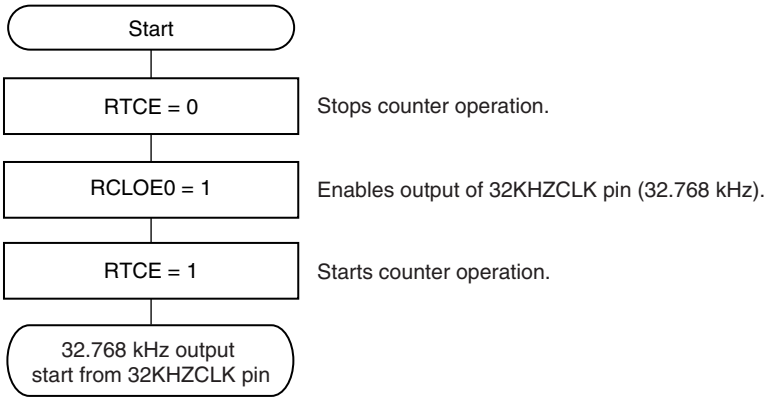
7.4.5 1 Hz output of real-time counter

Figure 7-24. 1 Hz Output Setting Procedure



7.4.6 32.768 kHz output of real-time counter

Figure 7-25. 32.768 kHz Output Setting Procedure



7.4.7 Example of watch error correction of real-time counter

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the watch error correction register.

Example of calculating the correction value

The correction value used when correcting the count value of the sub-count register (RSUBC) is calculated by using the following expression.

Set DEV to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(When DEV = 0)

Correction value[Note] = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60 ÷ 3

(When DEV = 1)

Correction value[Note] = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60

Note The correction value is the watch error correction value calculated by using bits 6 to 0 of the watch error correction register (SUBCUD).

(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) – 1} × 2

(When F6 = 1) Correction value = – {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), watch error correction is not performed. "*" is 0 or 1. /F5 to /F0 are bit-inverted values (000011 when 111100).

Remarks 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

2. The oscillation frequency is the input clock (f_{SUB}) value of the real-time counter (RTC).
It can be calculated from the 32 kHz output frequency of the 32KHZCLK pin or the output frequency of the RTC1HZ pin × 32768 when the watch error correction register is set to its initial value (00H).

3. The target frequency is the frequency resulting after correction performed by using the watch error correction register.

Correction example <1>

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32 kHz from the 32KHZCLK pin or outputting about 1 Hz from the RTC1HZ pin when the watch error correction register is set to its initial value (00H).

Note See **7.4.5 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **7.4.6 32.768 kHz output of real-time counter** for the setting procedure of outputting about 32 kHz from the 32KHZCLK pin.

[Calculating the correction value]

(When the output frequency from the 32KHZCLK pin is 32772.3 Hz)

If the target frequency is assumed to be 32768 Hz (32772.3 Hz – 131.2 ppm), the correction range for –131.2 ppm is –63.1 ppm or less, so assume DEV to be 0.

The expression for calculating the correction value when DEV is 0 is applied.

$$\begin{aligned}\text{Correction value} &= \text{Number of correction counts in 1 minute} \div 3\\ &= (\text{Oscillation frequency} \div \text{Target frequency} - 1) \times 32768 \times 60 \div 3\\ &= (32772.3 \div 32768 - 1) \times 32768 \times 60 \div 3\\ &= 86\end{aligned}$$

[Calculating the values to be set to (F6 to F0)]

(When the correction value is 86)

If the correction value is 0 or more (when delaying), assume F6 to be 0.

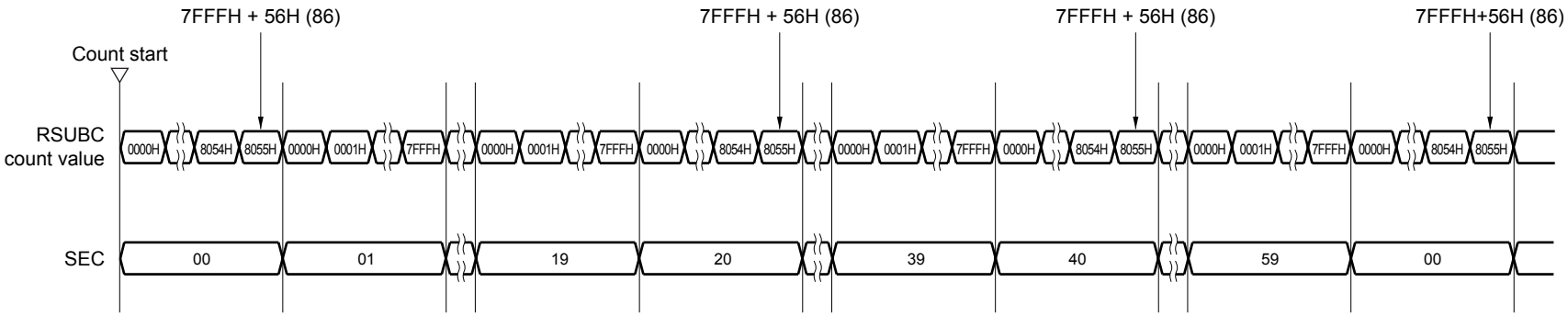
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

$$\begin{aligned}\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2 &= 86\\ (F5, F4, F3, F2, F1, F0) &= 44\\ (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 0, 0)\end{aligned}$$

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm), setting the correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of SUBCUD: 0101100) results in 32768 Hz (0 ppm).

Figure 7-26 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 7-26. Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)



Correction example <2>

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32 kHz from the 32KHZCLK pin or outputting about 1 Hz from the RTC1HZ pin when the watch error correction register is set to its initial value (00H).

Note See **7.4.5 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **7.4.6 32.768 kHz output of real-time counter** for the setting procedure of outputting about 32 kHz from the 32KHZCLK pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 0.9999817 Hz)

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

Correction value = Number of correction counts in 1 minute
= (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60
= (32767.4 ÷ 32768 − 1) × 32768 × 60
= −36

[Calculating the values to be set to (F6 to F0)]

(When the correction value is −36)

If the correction value is 0 or less (when speeding up), assume F6 to be 1.

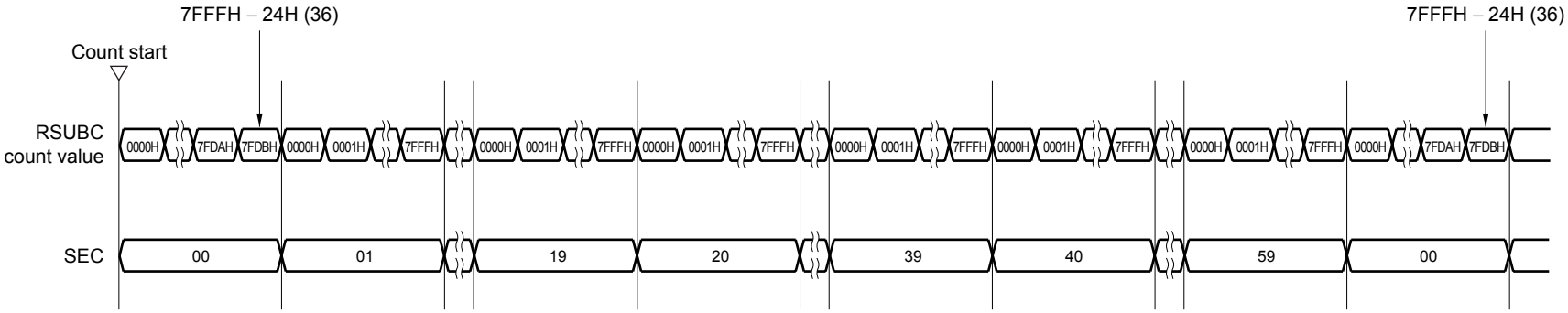
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

− {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2 = −36
(/F5, /F4, /F3, /F2, /F1, /F0) = 17
(/F5, /F4, /F3, /F2, /F1, /F0) = (0, 1, 0, 0, 0, 1)
(F5, F4, F3, F2, F1, F0) = (1, 0, 1, 1, 1, 0)

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is −36 (bits 6 to 0 of SUBCUD: 1101110) results in 32768 Hz (0 ppm).

Figure 7-27 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

Figure 7-27. Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)



CHAPTER 8 REAL-TIME COUNTER 2

8.1 Functions of Real-Time Counter 2

The real-time counter 2 has the following features.

- Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
- Constant-period interrupt function (period: 1 month to 0.5 seconds)
- Alarm interrupt function (alarm: week, hour, minute)
- Interval interrupt function

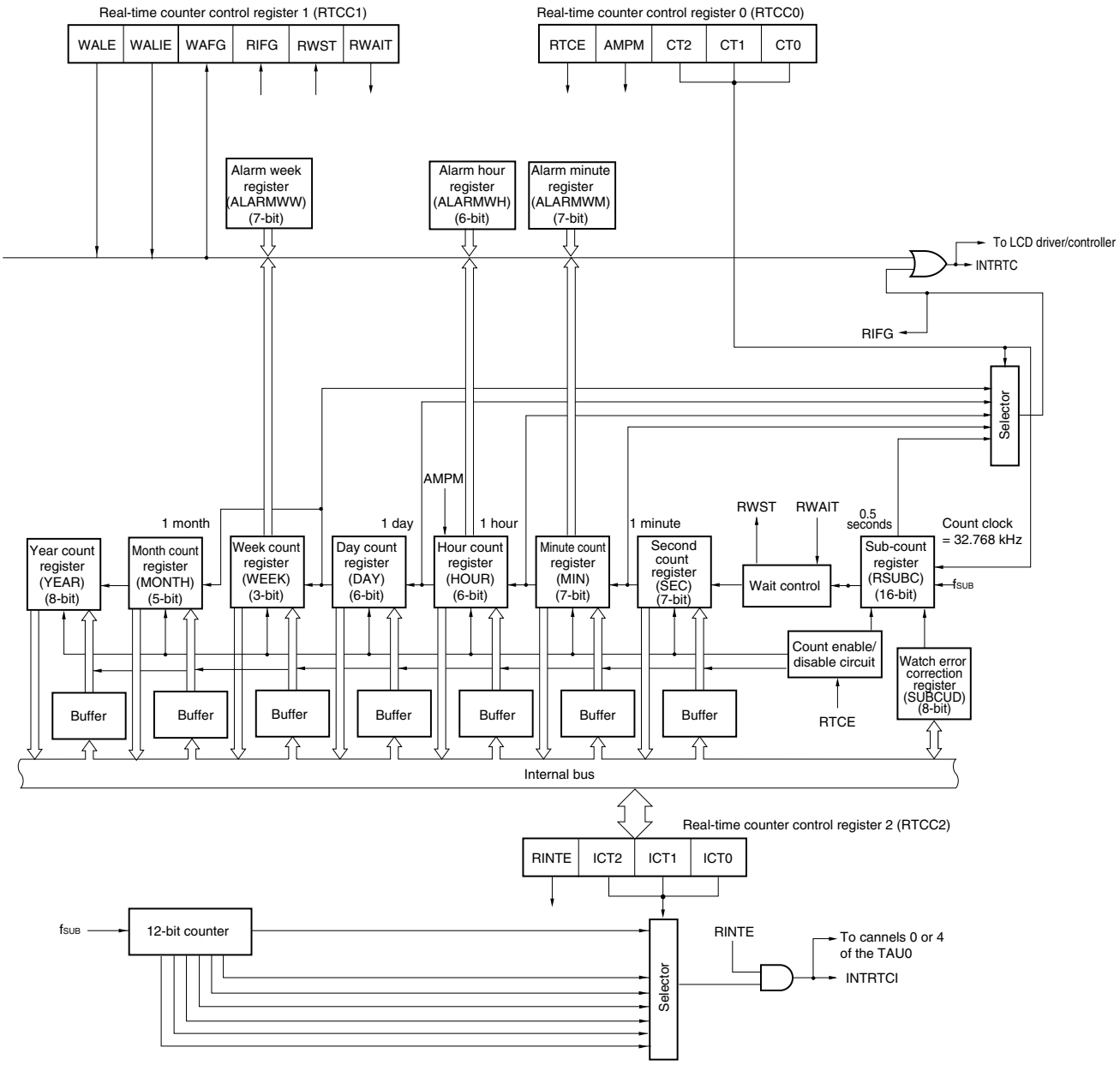
8.2 Configuration of Real-Time Counter 2

The real-time counter 2 includes the following hardware.

Table 8-1. Configuration of Real-Time Counter 2

Item	Configuration
Control registers	Peripheral enable register 0 (PER0)
	Real-time counter control register 0 (RTCC0)
	Real-time counter control register 1 (RTCC1)
	Real-time counter control register 2 (RTCC2)
	Sub-count register (RSUBC)
	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. Block Diagram of Real-Time Counter 2



8.3 Registers Controlling Real-Time Counter 2

Timer real-time counter 2 is controlled by the following 16 registers.

- Peripheral enable register 0 (PER0)
- Real-time counter control register 0 (RTCC0)
- Real-time counter control register 1 (RTCC1)
- Real-time counter control register 2 (RTCC2)
- Sub-count register (RSUBC)
- 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)

(1) Peripheral enable register 0 (PER0)

PER0 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 real-time counter 2 is used, be sure to set bit 7 (RTCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 8-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of real-time counter 2 (RTC2) input clock[Note]
0	Stops supply of input clock. • SFR used by the real-time counter 2 (RTC2) cannot be written. • The real-time counter 2 (RTC2) is in the reset status.
1	Supplies input clock. • SFR used by the real-time counter 2 (RTC2) can be read/written.

Note By using RTCEN, can supply and stop the clock that is used when accessing the real-time counter 2 (RTC2) from the CPU. RTCEN cannot control supply of the operating clock to RTC2.

Cautions 1. When using the real-time counter 2, first set RTCEN to 1, while oscillation of the subsystem clock (f_{SUB}) is stable. If RTCEN = 0, writing to a control register of the real-time counter 2 is ignored, and, even if the register is read, only the default value is read.

2. Clock supply to peripheral functions except the real-time counter 2 can be stopped in the HALT mode when operating on the subsystem clock by setting RTCLPC of the operation speed mode control register (OSMC) to 1. In this case, set RTCEN to 1 and bits 0 to 6 of PER0 to 0.

(2) Real-time counter control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time counter 2 operation, and set a 12- or 24-hour system and the constant-period interrupt function.

RTCC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-3. Format of Real-Time Counter Control Register 0 (RTCC0)

Address: FFF9DH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
RTCC0	RTCE	0	0	0	AMPM	CT2	CT1	CT0

RTCE	Real-time counter 2 operation control
0	Stops counter operation.
1	Starts counter operation.

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• To change the value of AMPM, set RWAIT (bit 0 of RTCC1) to 1, and re-set the hour count register (HOUR). • Table 8-2 shows the displayed time digits that are displayed.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
When changing the values of CT2 to CT0 while the counter operates (RTCE = 1), rewrite the values of CT2 to CT0 after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of CT2 to CT0, enable interrupt servicing after clearing the RIFG and RTCIF flags.			

Remark ×: don't care

(3) Real-time counter 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.

Reset signal generation clears this register to 00H.

Figure 8-4. Format of Real-Time Counter Control Register 1 (RTCC1) (1/2)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
When setting a value to the WALE bit while the counter operates (RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting each alarm register (WALIE flag of RTCC1, the ALARMWM register, the ALARMWH register, and the ALARMWW register), set match operation to be invalid (“0”) for the WALE bit.	

WALIE	Control of alarm interrupt (INTRTC) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to “1” one 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.	

Figure 8-4. Format of Real-Time Counter Control Register 1 (RTCC1) (2/2)

RIFG	Constant-period interrupt status flag
0	Constant-period interrupt is not generated.
1	Constant-period interrupt is generated.
This flag indicates the status of generation of the constant-period interrupt. When the constant-period interrupt is generated, it is set to “1”. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

RWST	Wait status flag of real-time counter 2
0	Counter is operating.
1	Mode to read or write counter value
This status flag indicates whether the setting of RWAIT is valid. Before reading or writing the counter value, confirm that the value of this flag is 1.	

RWAIT	Wait control of real-time counter 2
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value
This bit controls the operation of the counter. Be sure to write “1” to it to read or write the counter value. Because RSUBC continues operation, complete reading or writing of it in 1 second, and clear this bit back to 0. When RWAIT = 1, it takes up to 1 clock (32.768 kHz) until the counter value can be read or written. If RSUBC overflows when RWAIT = 1, it counts up after RWAIT = 0. If the second count register is written, however, it does not count up because RSUBC is cleared.	

Caution If writing is performed to the RTCC1 register with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to the RIFG and WAFG flags, be sure to use an 8-bit manipulation instruction. At this time, set 1 to the RIFG and WAFG flags to invalidate writing and not to clear the RIFG and WAFG flags during writing. When the value may be rewritten because the RIFG and WAFG flags are not being used, the RTCC1 register may be written by using a 1-bit manipulation instruction.

Remark Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

(4) Real-time counter control register 2 (RTCC2)

The RTCC2 register is an 8-bit register that is used to control the interval interrupt function.

RTCC2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-5. Format of Real-Time Counter Control Register 2 (RTCC2)

Address: FFF9FH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
RTCC2	RINTE	0	0	0	0	ICT2	ICT1	ICT0

RINTE	ICT2	ICT1	ICT0	Interval interrupt (INTRTCI) selection
0	×	×	×	Interval interrupt is not generated.
1	0	0	0	$2^6/f_{SUB}$ (1.953125 ms)
1	0	0	1	$2^7/f_{SUB}$ (3.90625 ms)
1	0	1	0	$2^8/f_{SUB}$ (7.8125 ms)
1	0	1	1	$2^9/f_{SUB}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{SUB}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{SUB}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{SUB}$ (125 ms)

Cautions 1. Change ICT2, ICT1, and ICT0 when RINTE = 0.

2. Be sure to clear bit 5 to “0”.

Remark f_{SUB}: Subsystem clock frequency

(5) Sub-count register (RSUBC)

The RSUBC register is a 16-bit register that counts the reference time of 1 second of the real-time counter 2. It takes a value of 0000H to 7FFFH and counts 1 second with a clock of 32.768 kHz.

RSUBC can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Cautions 1. When a correction is made by using the SUBCUD register, the value may become 8000H or more.

2. This register is also cleared by reset effected by writing the second count register.

3. The value read from this register is not guaranteed if it is read during operation, because a value that is changing is read.

Figure 8-6. Format of Sub-Count Register (RSUBC)

Address: FFF90H After reset: 0000H R

Symbol	7	6	5	4	3	2	1	0
RSUBC	SUBC7	SUBC6	SUBC5	SUBC4	SUBC3	SUBC2	SUBC1	SUBC0

Address: FFF91H After reset: 0000H R

Symbol	7	6	5	4	3	2	1	0
RSUBC	SUBC15	SUBC14	SUBC13	SUBC12	SUBC11	SUBC10	SUBC9	SUBC8

(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 counts up when the sub-counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later.

Set a decimal value of 00 to 59 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 8-7. Format of Second Count Register (SEC)

Address: FFF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

(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 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 (32.768 kHz) 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.

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

Reset signal generation clears this register to 00H.

Figure 8-8. Format of Minute Count Register (MIN)

Address: FFF93H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

(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 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 (32.768 kHz) 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. Set a decimal value of 00 to 23, or 01 to 12 and 21 to 32 to this register in BCD code.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Figure 8-9. Format of Hour Count Register (HOUR)

Address: FFF94H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution Bit 5 (HOUR20) of HOUR indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

Table 8-2. Displayed Time Digits

24-Hour Display (AMPM bit = 1)		12-Hour Display (AMPM bit = 0)	
Time	HOUR Register	Time	HOUR Register
0	00H	0 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	0 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.

(9) Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.
It counts up when the hour counter overflows.
This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) 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.
DAY can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 01H.

Figure 8-10. Format of Day Count Register (DAY)

Address: FFF96H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

(10) Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.
It counts up in synchronization with the day counter.
When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 06 to this register in BCD code.
WEEK can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 8-11. Format of Week Count Register (WEEK)

Address: FFF95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution The value corresponding to the month count register or the day count register is not stored in the week count register 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

(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 counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) 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. Set a decimal value of 01 to 12 to this register in BCD code.

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

Reset signal generation clears this register to 01H.

Figure 8-12. Format of Month Count Register (MONTH)

Address: FFF97H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

(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 counts up when the month counter 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 (32.768 kHz) later. Even if the month 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 99 to this register in BCD code. Set a decimal value of 00 to 99 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 8-13. Format of Year Count Register (YEAR)

Address: FFF98H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

(13) Watch error correction register (SUBCUD)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value (reference value: 7FFFH) that overflows from the sub-count register (RSUBC) to the second count register.

Rewrite the SUBCUD register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the SUBCUD register, enable interrupt servicing after clearing the interrupt request flag (RTCIF) and constant-period interrupt status flag (RIFG).

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

Reset signal generation clears this register to 00H.

Figure 8-14. Format of Watch Error Correction Register (SUBCUD)

Address: FFF99H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch error only when the second digits are at 00 (every 60 seconds).
Writing to the SUBCUD register at the following timing is prohibited. • When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H • When DEV = 1 is set: For a period of SEC = 00H	

F6	Setting of watch error correction value
0	Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.
When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected. * is 0 or 1. /F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100). Range of correction value: (when F6 = 0) 2, 4, 6, 8, … , 120, 122, 124 (when F6 = 1) –2, –4, –6, –8, … , –120, –122, –124	

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	–189.2 ppm to 189.2 ppm	–63.1 ppm to 63.1 ppm
Maximum excludes quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

Remark If a correctable range is –63.1 ppm or lower and 63.1 ppm or higher, set 0 to DEV.

(14) Alarm minute register (ALARMWM)

This register is used to set minutes of alarm.

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

Reset signal generation clears this register to 00H.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 8-15. Format of Alarm Minute Register (ALARMWM)

Address: FFF9AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

(15) Alarm hour register (ALARMWH)

This register is used to set hours of alarm.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 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.

Figure 8-16. Format of Alarm Hour Register (ALARMWH)

Address: FFF9BH After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution Bit 5 (WH20) of ALARMWH indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

(16) Alarm week register (ALARMWW)

This register is used to set date of alarm.

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

Reset signal generation clears this register to 00H.

Figure 8-17. Format of Alarm Week Register (ALARMWW)

Address: FFF9CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

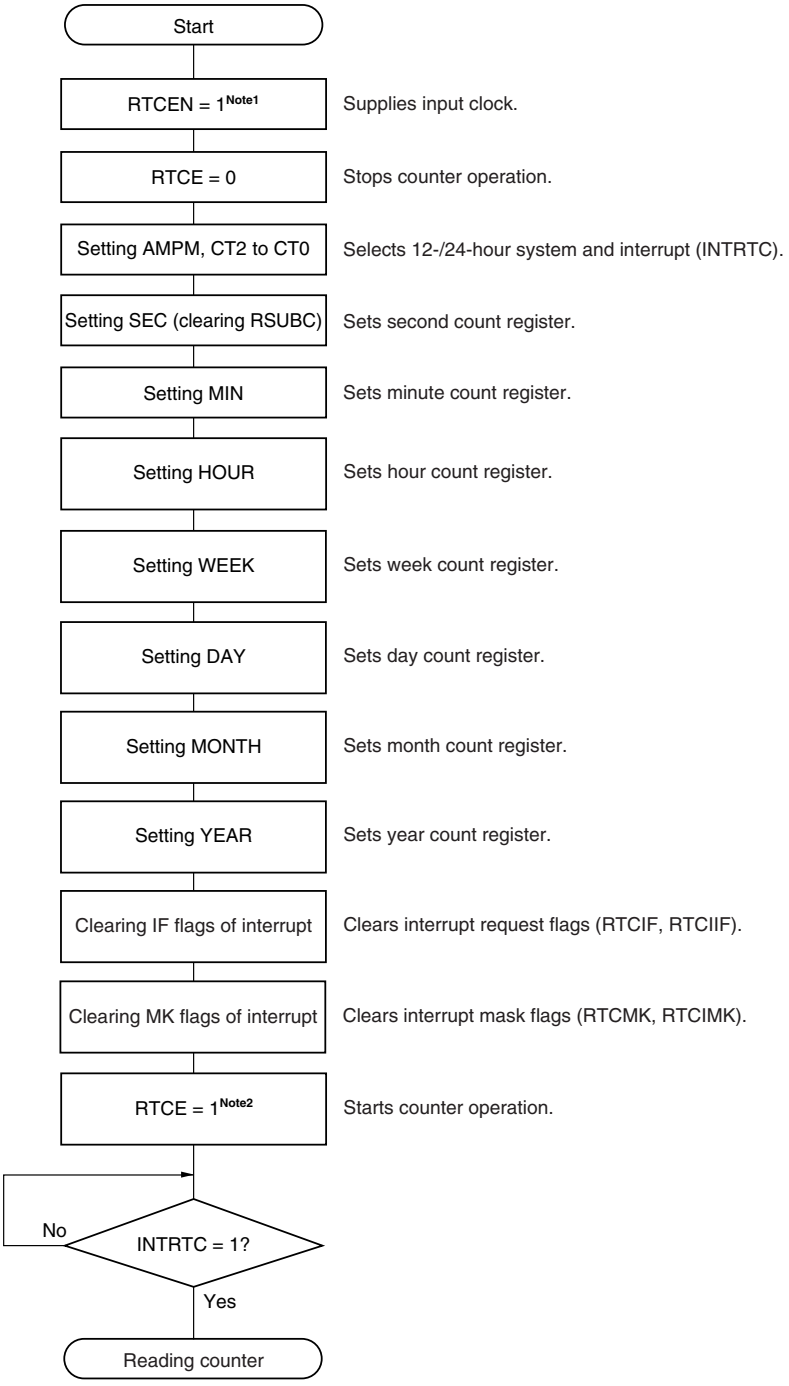
Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

8.4 Real-Time Counter 2 Operation

8.4.1 Starting operation of real-time counter 2

Figure 8-18. Procedure for Starting Operation of Real-Time Counter 2



Notes 1. First set RTCEN to 1, while oscillation of the subsystem clock (f_{SUB}) is stable.

2. Confirm the procedure described in **8.4.2 Shifting to STOP mode after starting operation** when shifting to STOP mode without waiting for INTRTC = 1 after RTCE = 1.

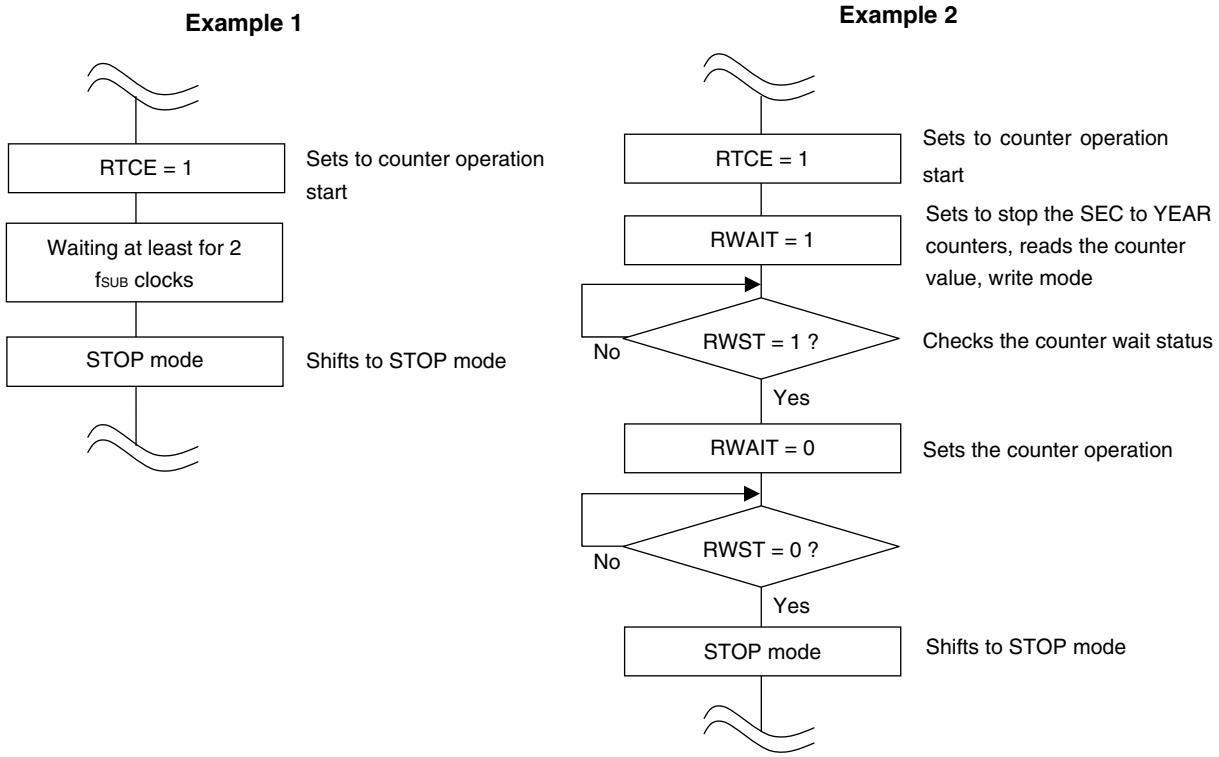
8.4.2 Shifting to STOP mode after starting operation

Perform one of the following processing when shifting to STOP mode immediately after setting RTCE to 1.

However, after setting RTCE to 1, this processing is not required when shifting to STOP mode after the first INTRTC interrupt has occurred.

- Shifting to STOP mode when at least two subsystem clocks (f_{SUB}) (about 62 μs) have elapsed after setting RTCE to 1 (see **Figure 8-19**, **Example 1**).
- Checking by polling RWST to become 1, after setting RTCE to 1 and then setting RWAIT to 1. Afterward, setting RWAIT to 0 and shifting to STOP mode after checking again by polling that RWST has become 0 (see **Figure 8-19**, **Example 2**).

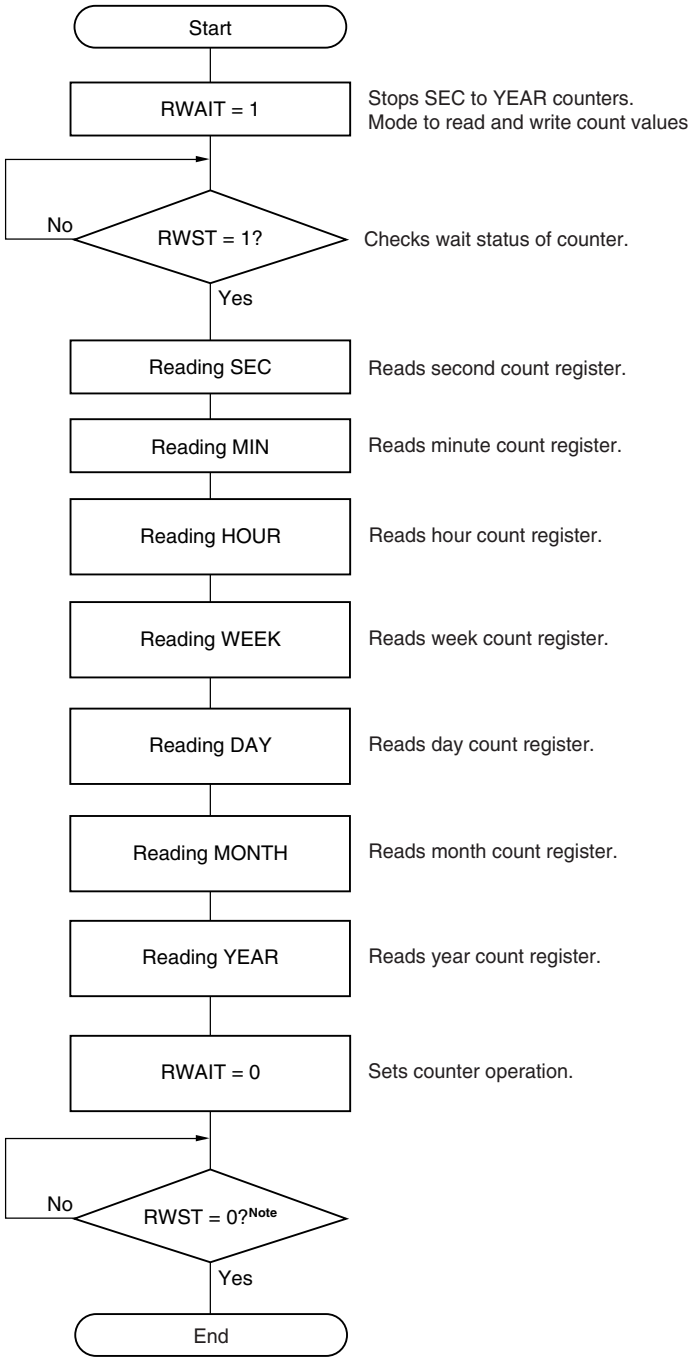
Figure 8-19. Procedure for Shifting to STOP Mode After Setting RTCE to 1



8.4.3 Reading/writing real-time counter 2

Read or write the counter after setting 1 to RWAIT first.

Figure 8-20. Procedure for Reading Real-Time Counter 2

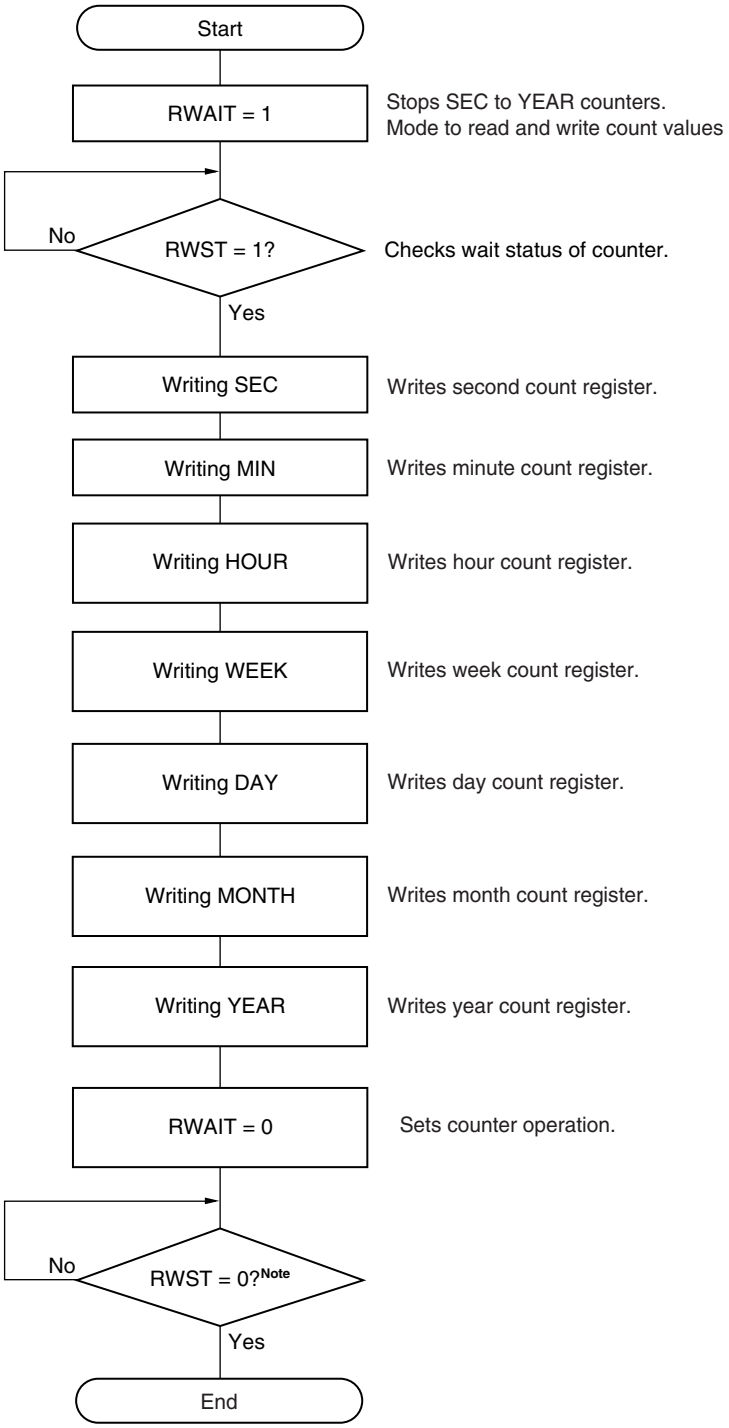


Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT 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.

Figure 8-21. Procedure for Writing Real-Time Counter 2



Note Be sure to confirm that RWST = 0 before setting STOP mode.

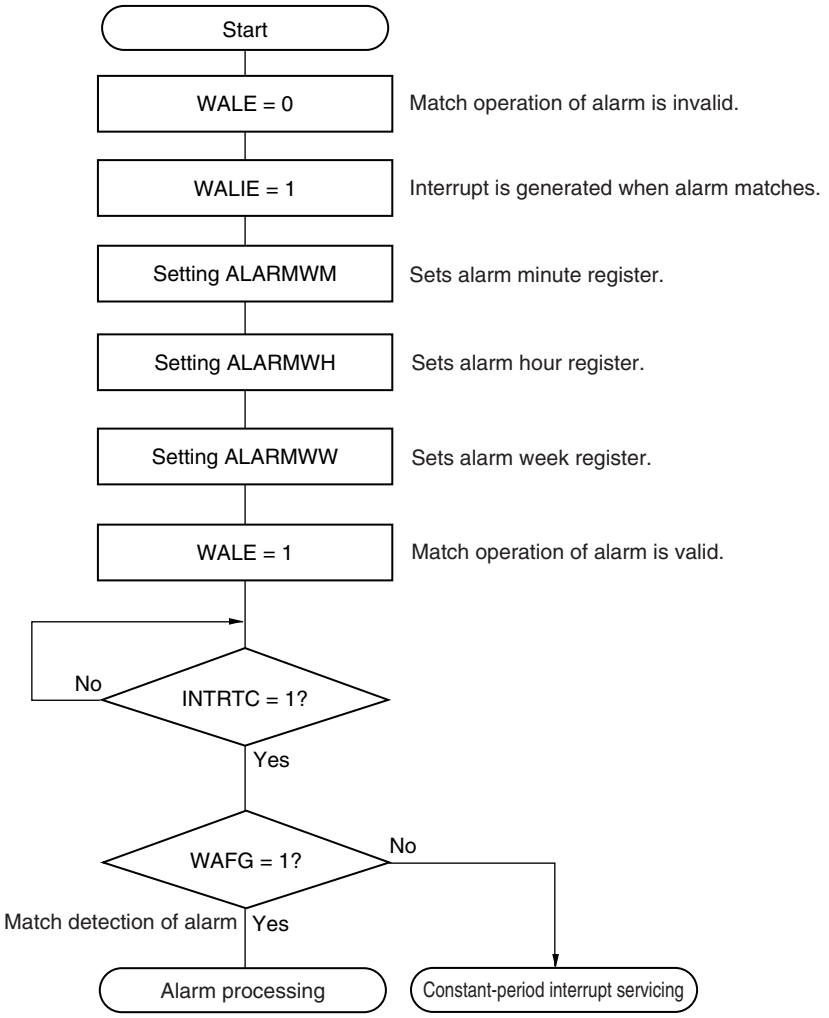
Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be written in any sequence.
All the registers do not have to be set and only some registers may be written.

8.4.4 Setting alarm of real-time counter 2

Set time of alarm after setting 0 to WALE first.

Figure 8-22. Alarm Setting Procedure



Remarks 1. ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

CHAPTER 9 WATCHDOG TIMER

9.1 Functions of Watchdog Timer

The watchdog timer is mounted onto all 78K0R/Lx3-M microcontroller products.

The watchdog timer operates on the internal low-speed oscillation 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 WDTE
- If data is written to WDTE during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDRF) of the reset control flag register (RESF) is set to 1. For details of RESF, see **CHAPTER 23 RESET FUNCTION**.

When 75% of the overflow time is reached, an interval interrupt can be generated.

9.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 9-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

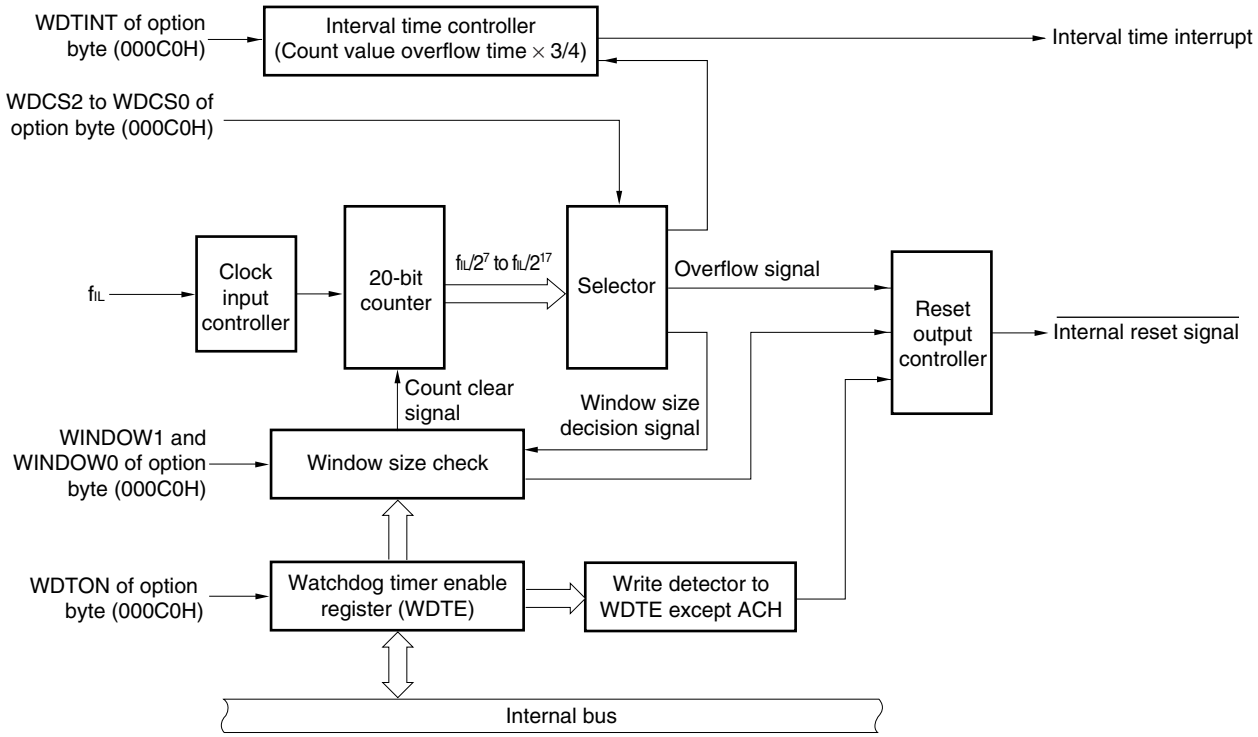
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 9-2. 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 9-1. Block Diagram of Watchdog Timer



9.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

(1) Watchdog timer enable register (WDTE)

Writing “ACH” to WDTE clears the watchdog timer counter and starts counting again.

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

Reset signal generation sets this register to 9AH or 1AH[Note].

Figure 9-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH After reset: 9AH/1AH[Note] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE reset value differs depending on the WDTON setting value of the option byte (000C0H). To operate watchdog timer, set WDTON to 1.

WDTON Setting Value	WDTE 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 WDTE, an internal reset signal is generated.

2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated.

3. The value read from WDTE is 9AH/1AH (this differs from the written value (ACH)).

9.4 Operation of Watchdog Timer

9.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 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 **9.4.2** 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 **9.4.3** and **CHAPTER 27**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to WDTE after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write WDTE the second time or later after a reset release during the window open period. If WDTE is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to WDTE, 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 watchdog timer enable register (WDTE)
   - If data other than “ACH” is written to WDTE

Cautions 1. When data is written to WDTE for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

2. If the watchdog timer is cleared by writing “ACH” to WDTE, the actual overflow time may be different from the overflow time set by the option byte by up to $2/f_{IL}$ seconds.

3. The watchdog timer can be cleared immediately before the count value overflows.

<Example> When the overflow time is set to $2^{10}/f_{IL}$, writing “ACH” is valid up to count value 3FH.

Cautions 4. The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP 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.

5. The watchdog timer continues its operation during self-programming of the flash memory and EEPROM® emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

9.4.2 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 WDTE during the window open period before the overflow time.

The following overflow time is set.

Table 9-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 33 kHz (MAX.))
0	0	0	$2^7/f_{IL}$ (3.88 ms)
0	0	1	$2^8/f_{IL}$ (7.76 ms)
0	1	0	$2^9/f_{IL}$ (15.52 ms)
0	1	1	$2^{10}/f_{IL}$ (31.03 ms)
1	0	0	$2^{12}/f_{IL}$ (124.12 ms)
1	0	1	$2^{14}/f_{IL}$ (496.48 ms)
1	1	0	$2^{15}/f_{IL}$ (992.97 ms)
1	1	1	$2^{17}/f_{IL}$ (3971.88 ms)

Caution The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

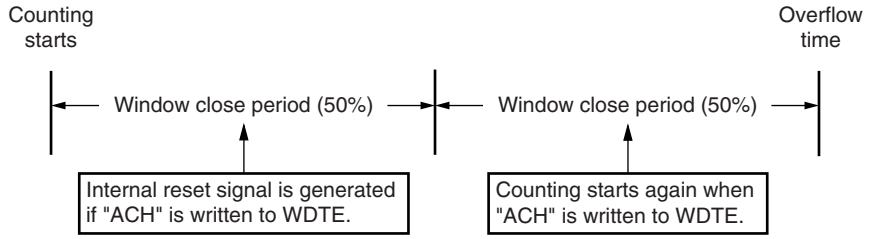
Remark f_{IL}: Internal low-speed oscillation clock frequency

9.4.3 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 WDTE during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to WDTE 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 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.

The window open period to be set is as follows.

Table 9-4. 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%

Cautions 1. The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

2. When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of WINDOW1 and WINDOW0.

Remark If the overflow time is set to $2^{10}/f_{IL}$, the window close time and open time are as follows.

($2.7\ V \leq V_{DD} \leq 3.6\ V$)

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 18.96 ms	0 to 9.48 ms	None
Window open time	18.96 to 31.03 ms	9.48 to 31.03 ms	0 to 31.03 ms

<When window open period is 50%>

- Overflow time:
  $2^{10}/f_{IL}$ (MAX.) = 2^{10}/33 kHz (MAX.) = 31.03 ms
- Window close time:
  0 to $2^{10}/f_{IL}$ (MIN.) × (1 − 0.5) = 0 to 2^{10}/27 kHz (MIN.) × 0.5 = 0 to 18.96 ms
- Window open time:
  $2^{10}/f_{IL}$ (MIN.) × (1 − 0.5) to $2^{10}/f_{IL}$ (MAX.) = 2^{10}/27 kHz (MIN.) × 0.5 to 2^{10}/33 kHz (MAX.) = 18.96 to 31.03 ms

9.4.4 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% of the overflow time is reached.

Table 9-5. 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% 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 WDTE register). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

CHAPTER 10 10-BIT SUCCESSIVE APPROXIMATION TYPE A/D CONVERTER

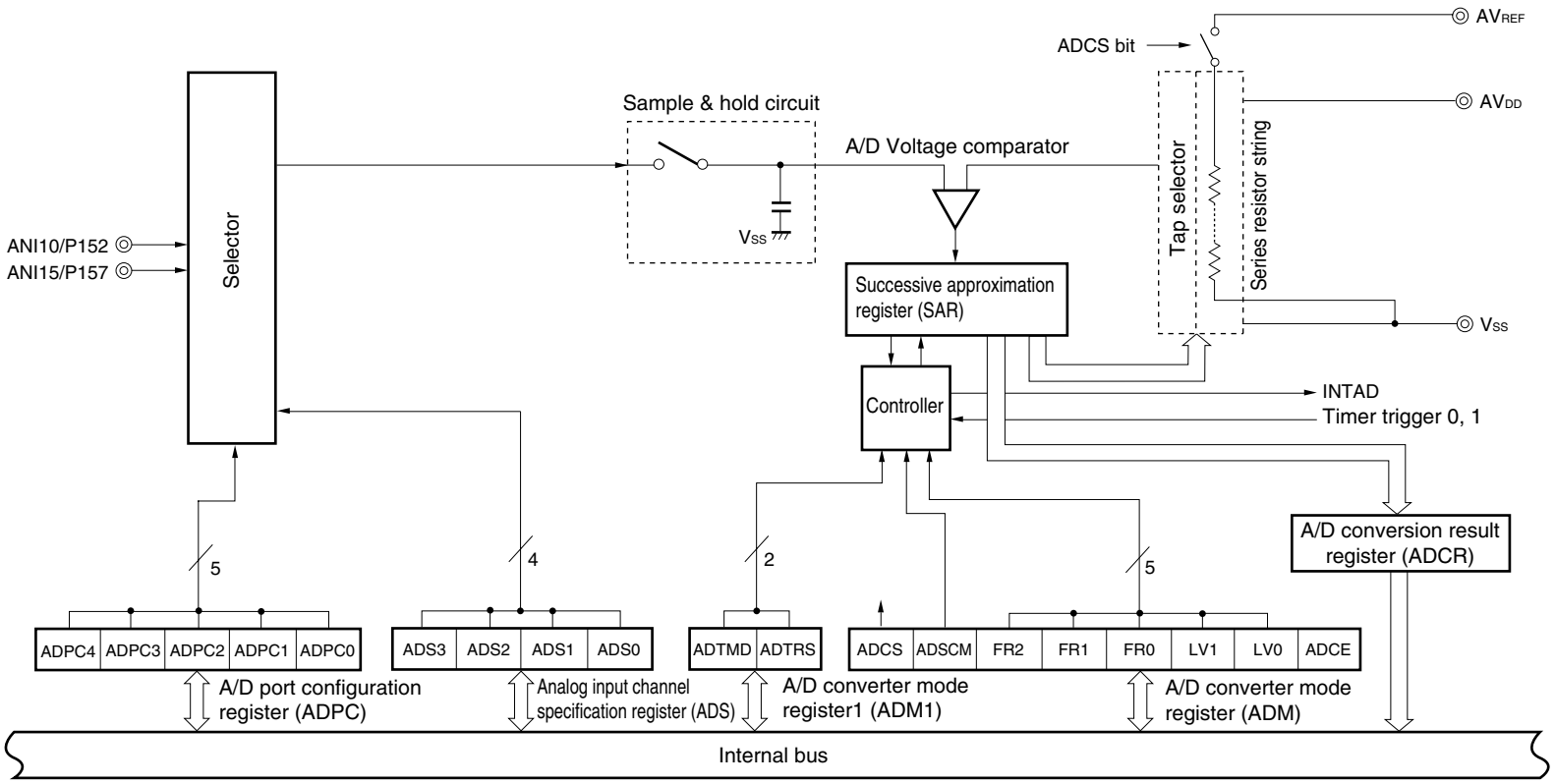
10.1 Function of 10-bit Successive Approximation Type A/D Converter

The A/D converter is a 10-bit resolution converter that converts analog input signals into digital values, and consists of two channels of A/D converter analog inputs (ANI10, ANI15).

The following four A/D converter operation modes are available.

- Software trigger mode (Continuous conversion mode)
- Software trigger mode (Single conversion mode)
- Timer trigger mode (Continuous conversion mode)
- Timer trigger mode (Single conversion mode)

Figure 10-1. Block Diagram of A/D Converter



10.2 Configuration of 10-bit Successive Approximation Type A/D Converter

The 10-bit successive approximation type A/D converter includes the following hardware.

(1) ANI10, ANI15 pins

These are the analog input pins of the 10-bit successive approximation type 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.

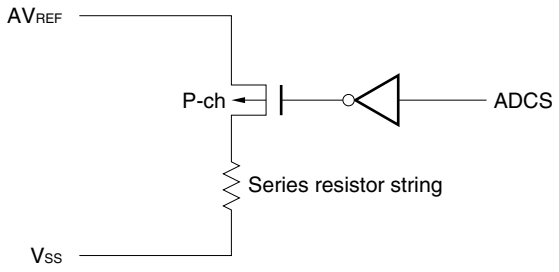
(2) Sample & hold circuit

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) Series resistor string

The series resistor string is connected between AV_{REF} and V_{SS}, and generates a voltage to be compared with the sampled voltage value.

Figure 10-2. Circuit Configuration of Series Resistor String



(4) Voltage comparator

The voltage comparator compares the sampled voltage value and the output voltage of the series resistor string.

(5) Successive approximation register (SAR)

This register converts the result of comparison by the voltage comparator, starting from the most significant bit (MSB). When the voltage value is converted into a digital value down to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred to the A/D conversion result register (ADCR).

(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 lower 10 bits (bits 2 to 11) (the higher 4 bits are fixed to 0 and lower 2 bits are undefined).

(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.

(9) AV_{DD} pin

This pin inputs an analog power to the A/D converter. When one or more of the pins of port 15 is used as the digital port pins, make AV_{DD} the same potential as V_{DD}.

(10) V_{SS} pin

This is the ground potential pin of the A/D converter.

(11) AV_{REF} pin

This pin is used to input the reference voltage of A/D converter.

The analog signal input to ANI10, ANI15 is converted into a digital signal, based on the voltage applied between AV_{REF} and V_{SS}.

10.3 Registers Used in 10-bit Successive Approximation Type A/D Converter

The 10-bit successive approximation type A/D converter uses the following eight registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register (ADM)
- A/D converter mode register 1 (ADM1)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- A/D port configuration register (ADPC)
- Port mode register 15 (PM15)

(1) Peripheral enable register 0 (PER0)

PER0 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 A/D converter is used, be sure to set bit 5 (ADCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 10-3. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

ADCEN	Control of 10-bit successive approximation type A/D converter input clock
0	Stops supply of input clock. • SFR used by the 10-bit successive approximation type A/D converter cannot be written. • The 10-bit successive approximation type A/D converter is in the reset status.
1	Supplies input clock. • SFR used by the 10-bit successive approximation type A/D converter can be read/written.

Caution When setting the 10-bit successive approximation type A/D converter, be sure to set ADCEN to 1 first. If ADCEN = 0, writing to a control register of the 10-bit successive approximation type A/D converter is ignored, and, even if the register is read, only the default value is read.

(2) A/D converter mode register (ADM)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

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

Reset signal generation clears this register to 00H.

Figure 10-4. Format of A/D Converter Mode Register (ADM)

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	ADSCM	FR2 Note 1	FR1 Note 1	FR0 Note 1	LV1 Note 1	LV0 Note 1	ADCE

ADCS	A/D conversion operation control Notes 2, 3, 4
0	Stops conversion operation
1	Enables conversion operation

ADSCM	A/D conversion operation mode specification
0	Continuous conversion mode
1	Single conversion mode

ADCE	A/D voltage comparator operation control Note 4
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

Notes 1. For details of FR2 to FR0, LV1, LV0, and A/D conversion, see **Table 10-2 A/D Conversion Time Selection (2.7 V ≤ AV_{REF} ≤ 3.6 V)**.

2. When using the A/D converter in timer trigger mode, do not set ADCS to 1. (ADCS automatically switches to 1 when a timer trigger signal is generated.) However, ADCS may be set to 0 to stop A/D conversion.

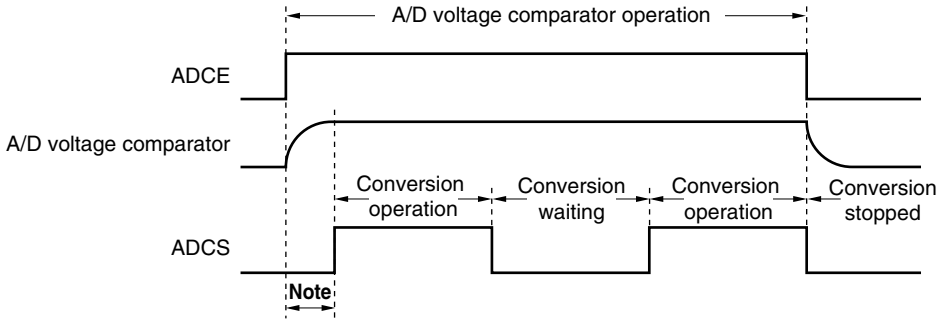
3. Read ADCS to determine whether A/D conversion is under execution.

4. The operation of the A/D voltage comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Table 10-1. Settings of ADCS and ADCE

ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion waiting mode (A/D voltage comparator operation, only comparator consumes power)
1	0	Setting prohibited
1	1	Conversion mode (A/D voltage comparator operation)

Figure 10-5. Timing Chart When A/D Voltage Comparator Is Used



Note To stabilize the internal circuit, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer.

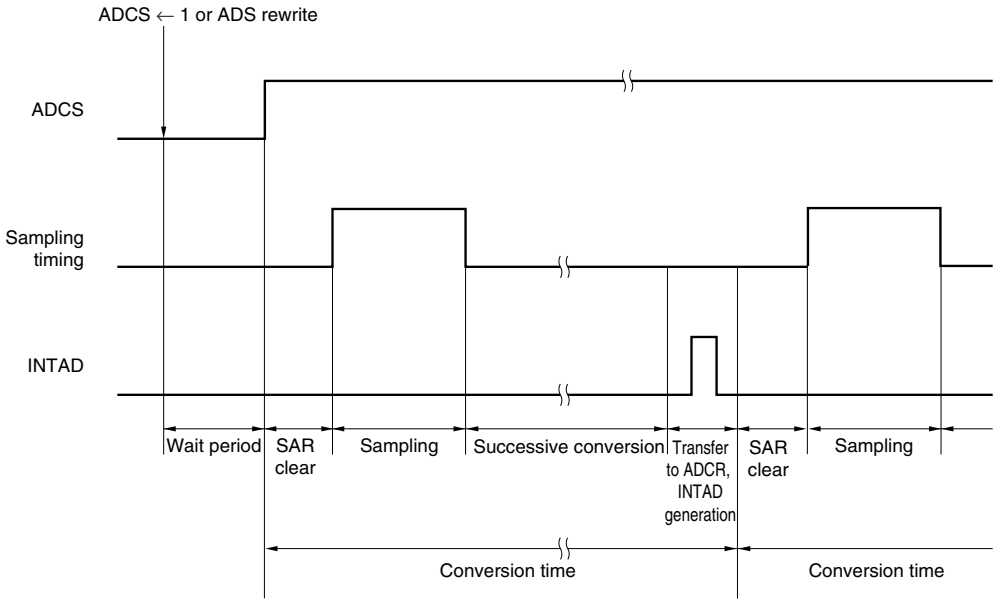
Caution A/D conversion must be stopped before rewriting bits ADSCM, FR0 to FR2, LV1, and LV0 to values other than the identical data.

Table 10-2. A/D Conversion Time Selection (2.7 V ≤ AV_{REF} ≤ 3.6 V)

A/D Converter Mode Register (ADM)					Conversion Time Selection					Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CLK} = 1 MHz	f_{CLK} = 8 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	
0	0	0	0	0	240/f_{CLK}	Setting prohibited	30 μs	24 μs	12 μs	f_{CLK}/12
0	0	1			160/f_{CLK}		20 μs	16 μs	8 μs	f_{CLK}/8
0	1	0			120/f_{CLK}		15 μs	12 μs	6 μs	f_{CLK}/6
0	1	1			100/f_{CLK}		12.5 μs	10 μs	5 μs	f_{CLK}/5
1	0	0			80/f_{CLK}		10 μs	8 μs	Setting prohibited	f_{CLK}/4
1	0	1			60/f_{CLK}		7.5 μs	6 μs		f_{CLK}/3
1	1	0			40/f_{CLK}	40 μs	5 μs	Setting prohibited		f_{CLK}/2
1	1	1			20/f_{CLK}	20 μs	Setting prohibited			f_{CLK}
Other than above					Setting prohibited					

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 10-6. A/D Converter Sampling and A/D Conversion Timing



(3) A/D converter mode register 1 (ADM1)

This register sets the A/D conversion start trigger.

ADM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-7. 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	ADTMD	0	0	0	0	0	0	ADTRS

ADTMD	A/D trigger mode selection
0	Software trigger mode
1	Timer trigger mode (hardware trigger mode)

ADTRS	Timer trigger signal selection
0	INTTM02
1	INTTM03

Caution Rewriting ADM1 during A/D conversion is prohibited. Rewrite it when conversion operation is stopped (ADCS = 0).

(4) 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result in the select mode. The higher 4 bits are fixed to 0 and lower 2 bits are undefined. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register. The higher 4 bits of the conversion result are stored in FFF1FH and the lower 8 bits are stored in the FFF1EH.

ADCR can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 10-8. Format of 10-bit A/D Conversion Result Register (ADCR)

Address: FFF1EH, FFF1FH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR	0	0	0	0											Un defined	Un defined

Caution When writing to A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.

(5) 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.

ADCRH can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-9. Format of 8-bit A/D Conversion Result Register (ADCRH)

Address: FFF1FH[Note] After reset: 00H R

Symbol	FFF1FH								FFF1EH							
ADCRH	0	0	0	0											Un defined	Un defined

ADCRH

Note If address FFF1FH is read, the data of ADCRH (lower four bits of FFF1FH and higher four bits of FFF1EH) will be read.

Caution When writing to A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.

(6) Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

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

Reset signal generation clears this register to 00H.

Figure 10-10. Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

ADS3	ADS2	ADS1	ADS0	Analog input channel
1	0	1	0	ANI10
1	1	1	1	ANI15
Other than the above				Setting prohibited

Cautions 1. Be sure to clear bits 4 to 7 to "0".

2 Set a channel to be used for A/D conversion in the input mode by using port mode register 15 (PM15).

3. Do not set the pin that is set by ADPC as digital I/O by ADS.

(7) A/D port configuration register (ADPC)

This register switches the ANI10/P152 and ANI15/P157 pins to analog input of A/D converter or digital I/O of port.

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

Reset signal generation sets this register to 10H.

Figure 10-11. Format of A/D Port Configuration Register (ADPC)

Address: F0017H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching	
					Port 15	
					ANI15/P157	ANI10/P152
0	1	0	1	0	A	A
0	1	1	1	1	A	D
1	0	0	0	0	D	D
Other than the above					Setting prohibited	

Cautions 1. Set a channel to be used for A/D conversion in the input mode by using port mode register 15 (PM15).

2. Do not set the pin that is set by ADPC as digital I/O by ADS.

(8) Port mode register 15 (PM15)

When using ANI10/P152 and ANI15/P157 pins for analog input port, set PM152, and P157 to 1. The output latches of P152 and P157 at this time may be 0 or 1.

If PM152 and PM157 are set to 0, they cannot be used as analog input port pins.

PM15 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Caution If a pin is set as an analog input port, not the pin level but "0" is always read.

Figure 10-12. Formats of Port Mode Register 15 (PM15)

Address: FFF2FH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM15	PM157	1	1	1	1	PM152	PM151[Note]	PM150[Note]

PMmn	Pmn pin I/O mode selection (mn = 152, 157)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note Be sure to clear PM150 and PM151 bits of PM15 to "0" after resetting.

The ANI10/P152 and ANI15/P157 pins are as shown below depending on the settings of ADPC register, ADS register, and PM15 register.

Table 10-3. Setting Functions of ANI10/P152 and ANI15/AM157 Pins

ADPC register	PM15 register	ADS register	ANI10/P152 and ANI15/AM157 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be A/D converted)
		Does not select ANI.	Analog input (not to be A/D converted)
	Output mode	–	Setting prohibited

10.4 A/D Converter Operations

10.4.1 Basic operations of A/D converter

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1 to start the supply of the input clock to the A/D converter.

<2> Set the A/D conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of A/D converter mode register (ADM), and set the operation mode by using bit 6 (ADMD) of ADM.

<3> Set bit 0 (ADCE) of ADM to 1 to start the operation of the A/D voltage comparator.

<4> Set the channels for A/D conversion to analog input by using the A/D port configuration register (ADPC) and set to input mode by using port mode registers (PM15).

<5> Select one channel for A/D conversion using the analog input channel specification register (ADS).

<6> Use the A/D converter mode register 1 (ADM1) to set the trigger mode.

<7> Start the conversion operation by setting bit 7 (ADCS) of ADM to 1, if the software trigger mode has been set in step <6>.

If timer trigger mode was specified in step <6>, ADCS is automatically set to 1 and A/D conversion starts when the timer trigger signal is detected. (<8> to <14> are operations performed by hardware.)

<8> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<9> 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.

<10> 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.

<11> 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 of SAR remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset to 0.

<12> Next, bit 8 of SAR is automatically set, 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 SAR is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<13> Comparison is continued in this way up to bit 0 of SAR.

<14> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.

At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<15> If single conversion mode has been set in step <2>, ADCS is automatically cleared to 0 and enters a wait state after the first A/D conversion ends.

If the continuous conversion mode has been set in step <2>, repeat steps <8> to <14>. To stop the A/D converter, clear ADCS to 0.

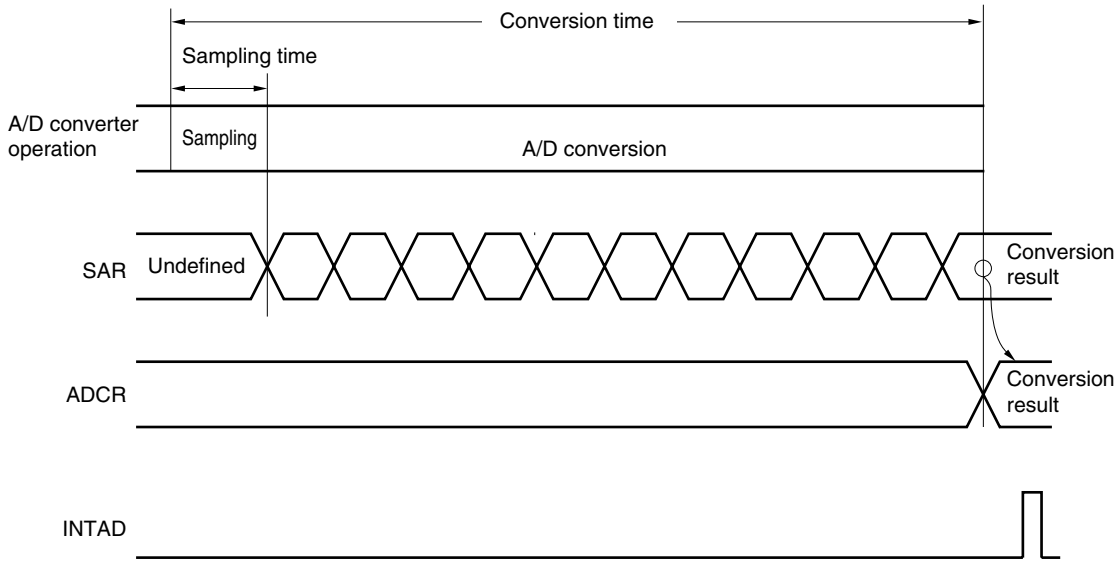
To restart A/D conversion from the status of ADCE = 1, start from <7>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start step <7>. To change the channel to be A/D converted, perform step <5>.

Caution Make sure the period of <3> to <7> is 1 μs or more.

Remark Two types of A/D conversion result registers are available. Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

- ADCR (16 bits): Store 10-bit A/D conversion value
- ADCRH (8 bits): Store 8-bit A/D conversion value

Figure 10-13. Basic Operation of A/D Converter



10.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI10, ANI15) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

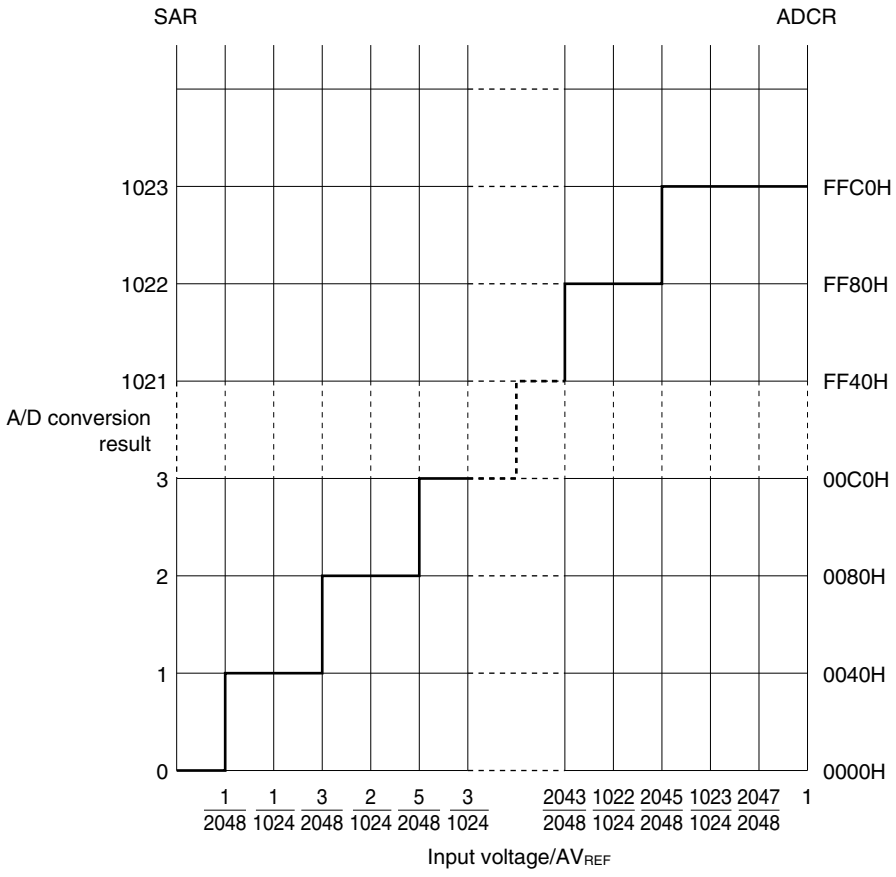
AV_{REF}: Reference voltage of A/D converter

ADCR: 10-bit A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 10-14 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 10-14. Relationship Between Analog Input Voltage and A/D Conversion Result



10.4.3 A/D converter operation modes

The following four A/D converter operation modes are available.

- Software trigger mode (Continuous conversion mode)
- Software trigger mode (Single conversion mode)
- Timer trigger mode (Continuous conversion mode)
- Timer trigger mode (Single conversion mode)

(1) Software trigger mode (Continuous conversion mode)

<1> By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

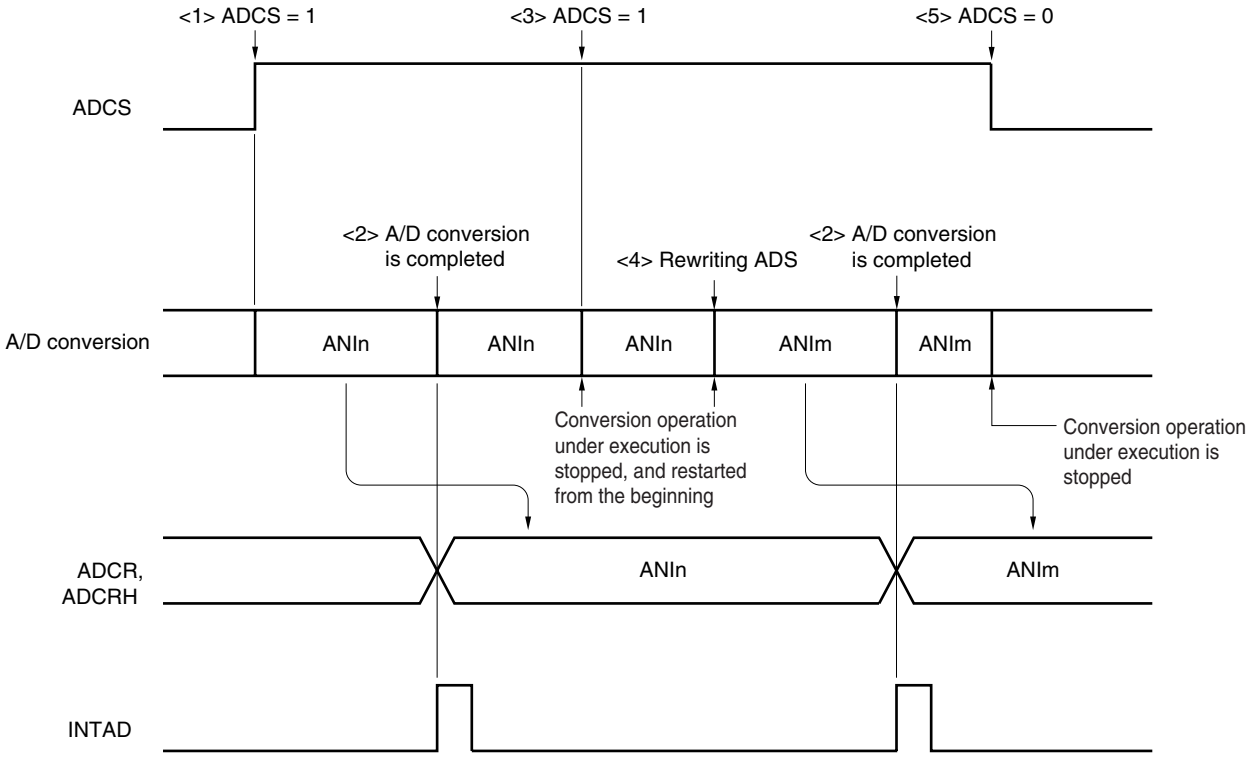
<2> When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, the next A/D conversion operation is immediately started.

<3> If 1 is written to ADCS during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<4> If ADS is rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<5> If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped. At this time, the conversion result immediately before is retained.

Figure 10-15. Software trigger mode (Continuous conversion mode)



Remark n = 10, 15
m = 10, 15

(2) Software trigger mode (Single conversion mode)

<1> By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

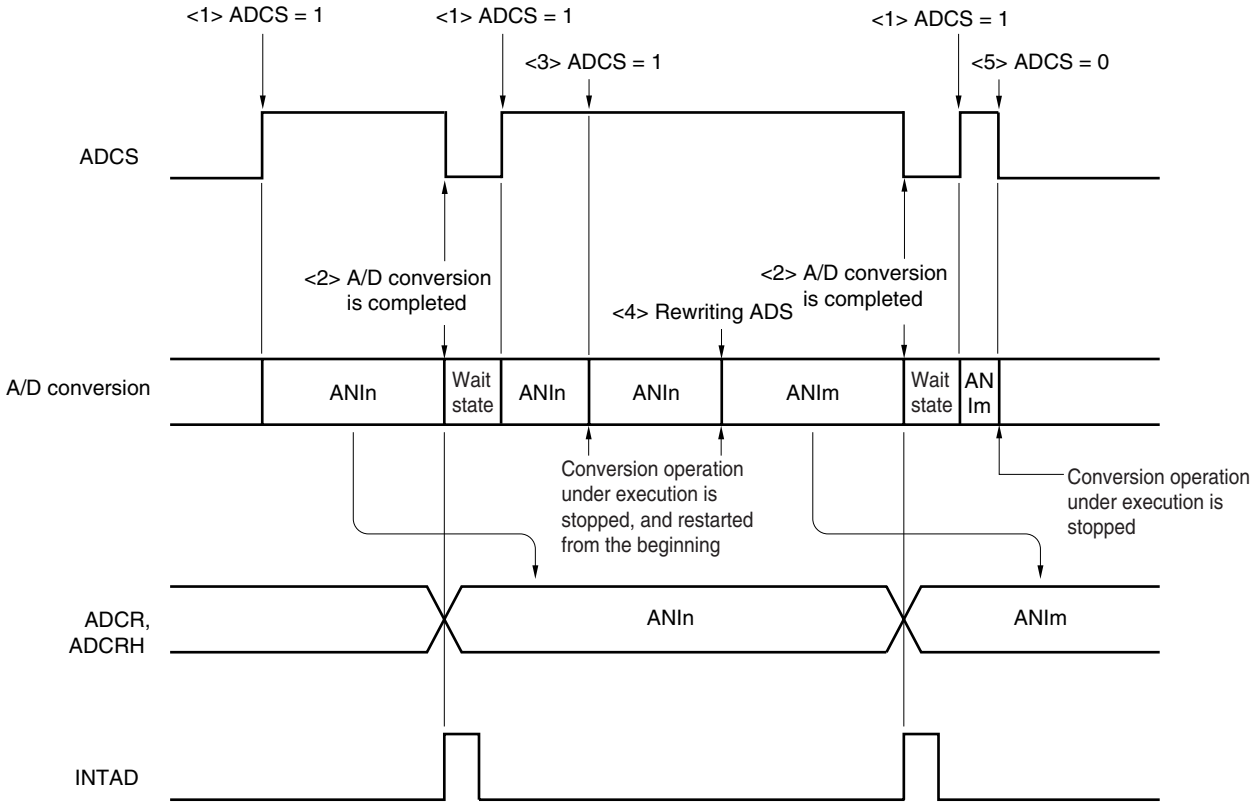
<2> When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, ADCS is automatically cleared and an A/D conversion wait state is entered.

<3> If 1 is written to ADCS during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<4> If ADS is rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<5> If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped. At this time, the conversion result immediately before is retained.

Figure 10-16. Software trigger mode (Single conversion mode)



Remark n = 10, 15
m = 10, 15

(3) Timer trigger mode (Continuous conversion mode)

<1> Timer trigger mode is set and a timer trigger wait state is entered by setting bit 7 (ADTMD) of A/D converter mode register 1 (ADM1) to 1.

<2> When the timer trigger signal is detected, bit 7 (ADCS) of the A/D converter mode register (ADM) is automatically set to 1 and A/D conversion of the voltage applied to the analog input pin specified using the analog input channel specification register (ADS) starts.

<3> When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, the next A/D conversion operation is immediately started.

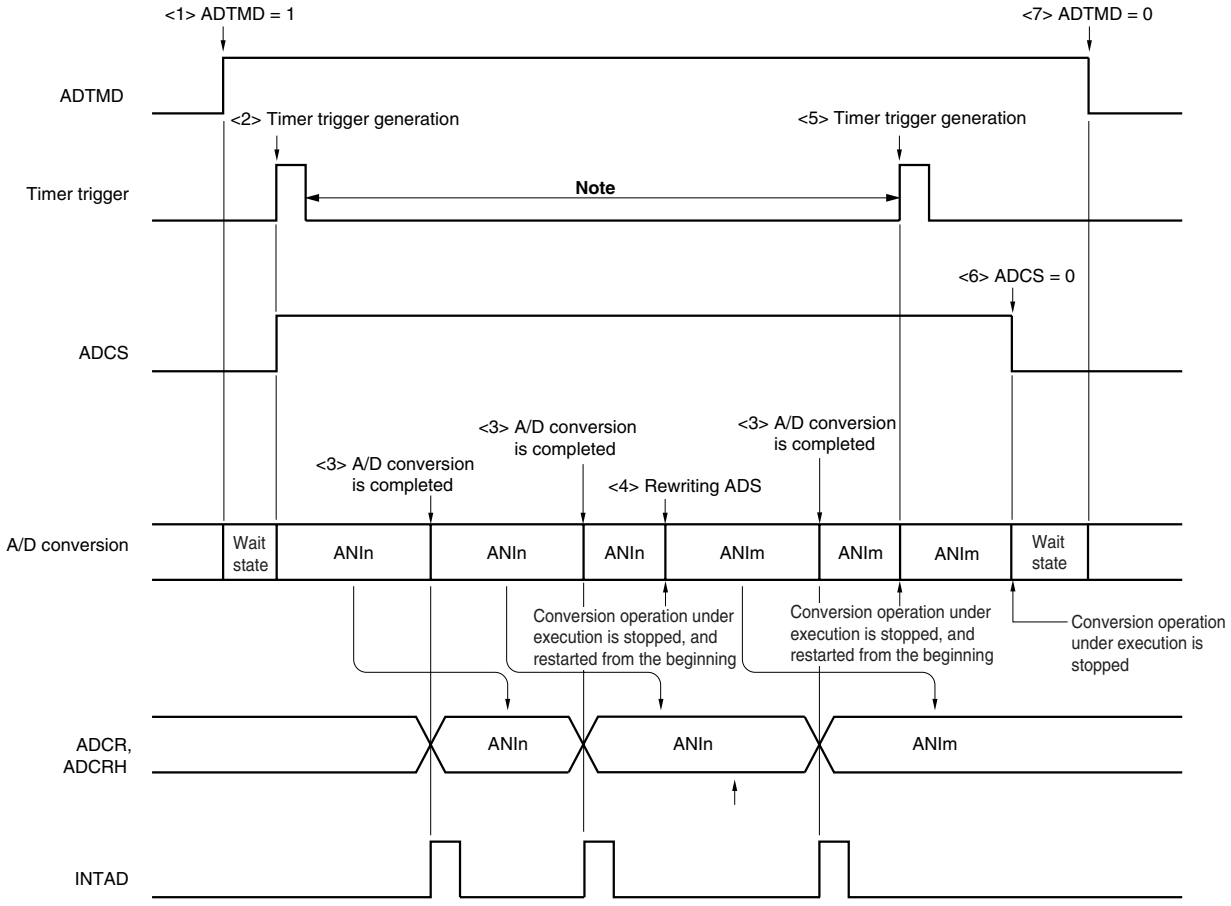
<4> If 1 is written to ADS during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<5> If a timer trigger signal is generated during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<6> If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped, and a timer trigger wait state is entered. At this time, the conversion result immediately before is retained.

<7> When 0 is written to ADTMD while A/D conversion operation is stopped (ADCS = 0), the software trigger mode is set and A/D conversion operation is not started, even if a timer trigger signal is generated.

Figure 10-17. Timer trigger mode (Continuous conversion mode)



Note Leave at least enough time for A/D conversion to finish between each generation of the timer trigger signal.

Remark n = 10, 15
m = 10, 15

(4) Timer trigger mode (Single conversion mode)

<1> Timer trigger mode is set and a timer trigger wait state is entered by setting bit 7 (ADTMD) of A/D converter mode register 1 (ADM1) to 1.

<2> When the timer trigger signal is detected, bit 7 (ADCS) of the A/D converter mode register (ADM) is automatically set to 1 and A/D conversion of the voltage applied to the analog input pin specified using the analog input channel specification register (ADS) starts.

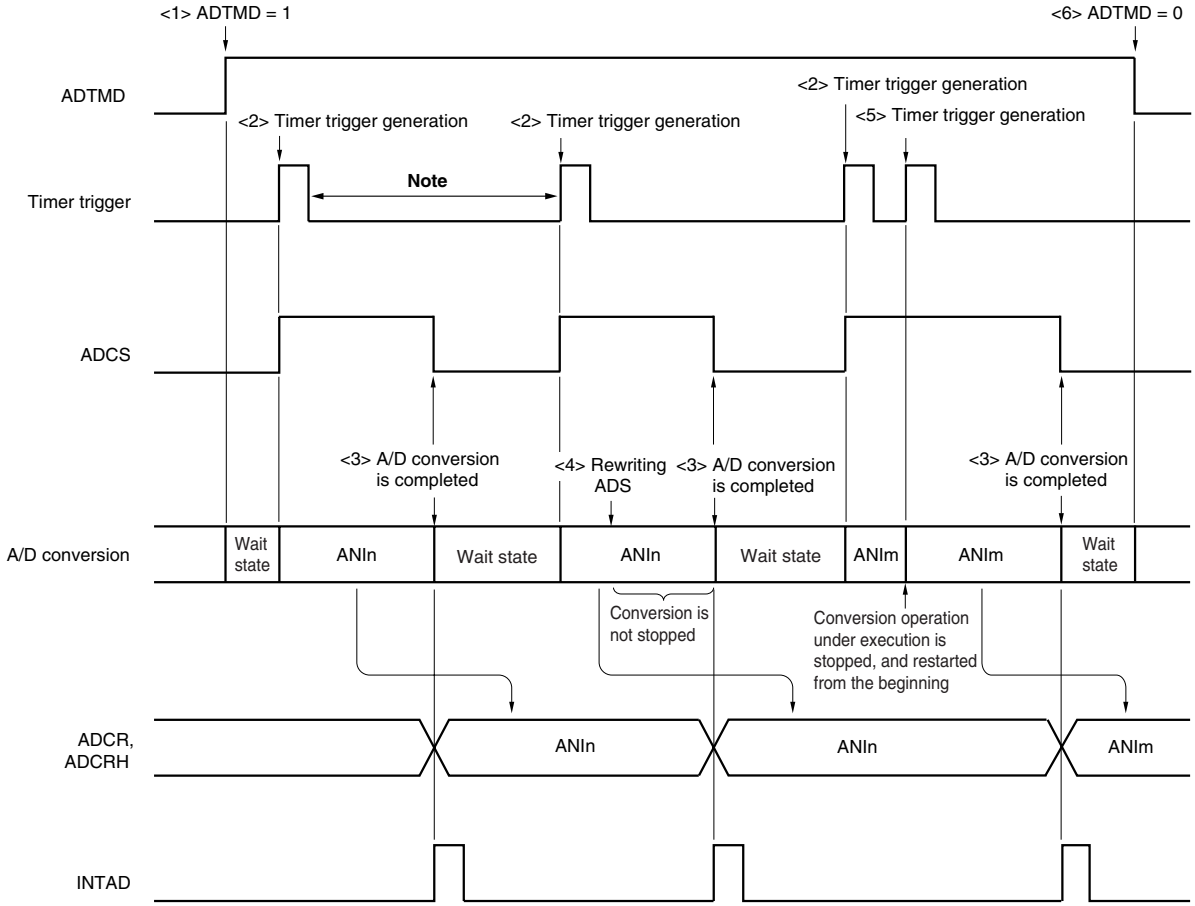
<3> When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, ADCS is automatically cleared and a timer trigger wait state is entered.

<4> Even if ADS is rewritten during an A/D conversion operation, the A/D conversion operation performed at that time is continued. The channel will be switched when the next A/D conversion operation starts.

<5> If a timer trigger signal is generated during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<6> When 0 is written to ADTMD while A/D conversion operation is stopped (ADCS = 0), the software trigger mode is set and A/D conversion operation is not started, even if a timer trigger signal is generated.

Figure 10-18. Timer trigger mode (Single conversion mode)



Note Leave at least enough time for A/D conversion to finish between each generation of the timer trigger signal.

Remark n = 10, 15
m = 10, 15

The setting methods are described below.

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1.
<2> Select the conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of A/D converter mode register (ADM), and select the operation mode by using bit 6 (ADSCM) of ADM.
<3> Set bit 0 (ADCE) of ADM to 1.
<4> Set the channel to be used in the analog input mode by using bits 4 to 0 (ADPC4 to ADPC0) of the A/D port configuration register (ADPC), bits 7, 2 to 0 (PM157, PM152) of port mode register 15 (PM15).
<5> Select a channel to be used by using bits 3 to 0 (ADS3 to ADS0) of the analog input channel specification register (ADS).
<6> Use bits 0 and 7 (ADTRS, ADTMD) of A/D converter mode register 1 (ADM1) to set the trigger mode.
<7> In the software trigger mode
→ Start A/D conversion by setting bit 7 (ADCS) of ADM to 1.
In the timer trigger mode
→ ADCS is automatically set to 1 and A/D conversion starts when the timer trigger signal is generated.
<8> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<9> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).
<10> In the continuous conversion mode
→ Start the next A/D conversion automatically.
In the single conversion mode
→ ADCS is automatically cleared to 0 and the A/D converter goes on standby. To start A/D conversion operation, go to step <7>.

<Change the channel>
<11> Change the channel using bits 3 to 0 (ADS3 to ADS0) of ADS to start A/D conversion.[Note]
<12> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<13> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Complete A/D conversion>
<14> Clear ADCS to 0.
<15> In the software trigger mode
→ Clear ADCE to 0.
In the timer trigger mode
→ Clear ADCE and ADTMD to 0.
<16> Clear bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 0.

Note When in timer trigger mode (single conversion mode), the A/D conversion operation is continued even if bits 3 to 0 of ADS are set during A/D conversion. The channel will be changed when the next A/D conversion operation starts.
When in any other mode, A/D conversion operation is aborted after bits 3 to 0 of ADS have been set, and A/D conversion operation is started from the beginning after the channel has been changed.

Cautions 1. Make sure the period of <3> to <7> is 1 μs or more.
2. <3> may be done between <4> and <6>.
3. <3> can be omitted. However, ignore data of the first conversion after <8> in this case.
4. The period from <8> to <12> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM. The period from <11> to <12> is the conversion time set using FR2 to FR0, LV1, and LV0.

10.5 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.
Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 10-19. Overall Error

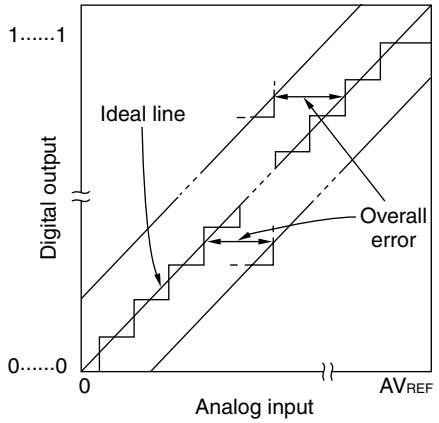
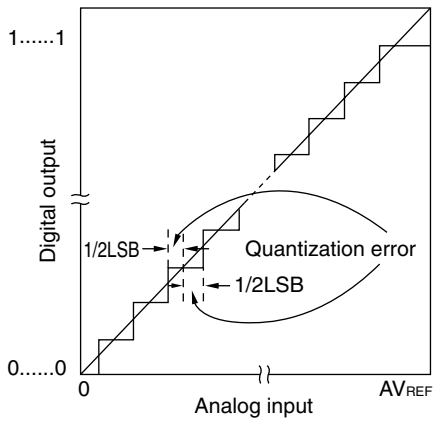


Figure 10-20. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.
If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 10-21. Zero-Scale Error

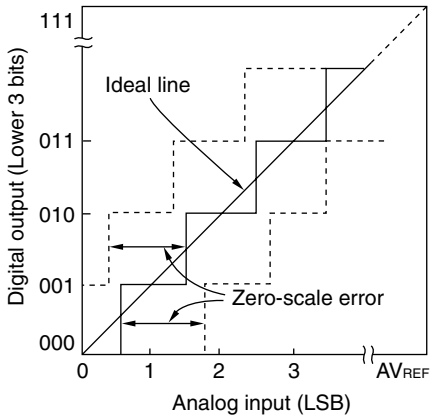


Figure 10-22. Full-Scale Error

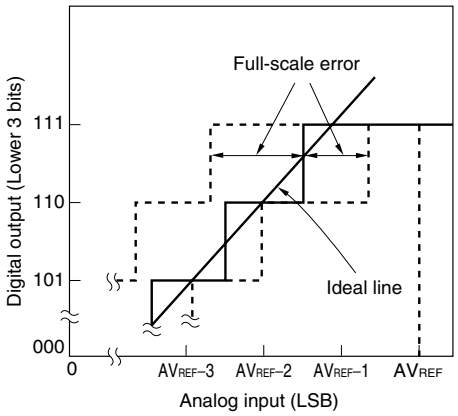


Figure 10-23. Integral Linearity Error

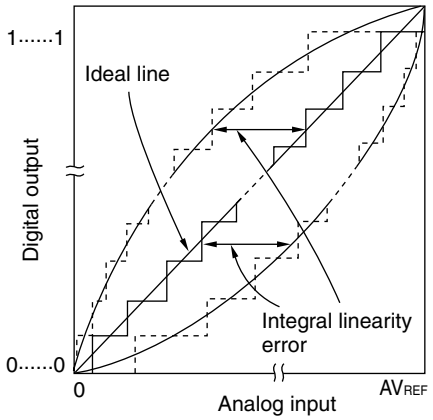
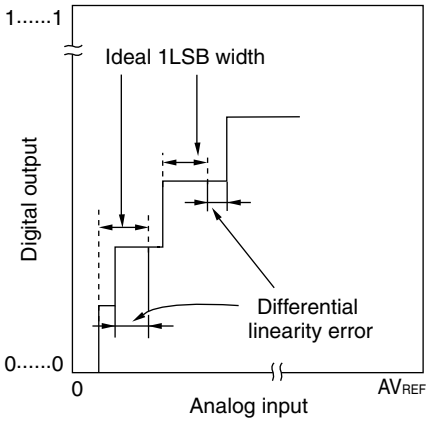


Figure 10-24. Differential Linearity Error



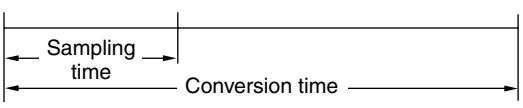
(8) Conversion time

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.



10.6 Cautions for 10-bit Successive Approximation Type 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 the A/D converter mode register (ADM) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1L (IF1L) to 0 and start operation.

(2) Input range of ANI10, ANI15

Observe the rated range of the ANI10, ANI15 input voltage. If a voltage of AV_{REF} or higher and V_{SS} or lower (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

(3) Conflicting operations

<1> Conflict between A/D conversion result register (ADCR, ADCRH) write and ADCR or ADCRH read by instruction upon the end of conversion

ADCR or ADCRH read has priority. After the read operation, the new conversion result is written to ADCR or ADCRH.

<2> Conflict between ADCR or ADCRH write and A/D converter mode register (ADM) write, analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion

ADM, ADS, or ADPC write has priority. ADCR or ADCRH 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_{REF}, ANI10, and ANI15 pins.

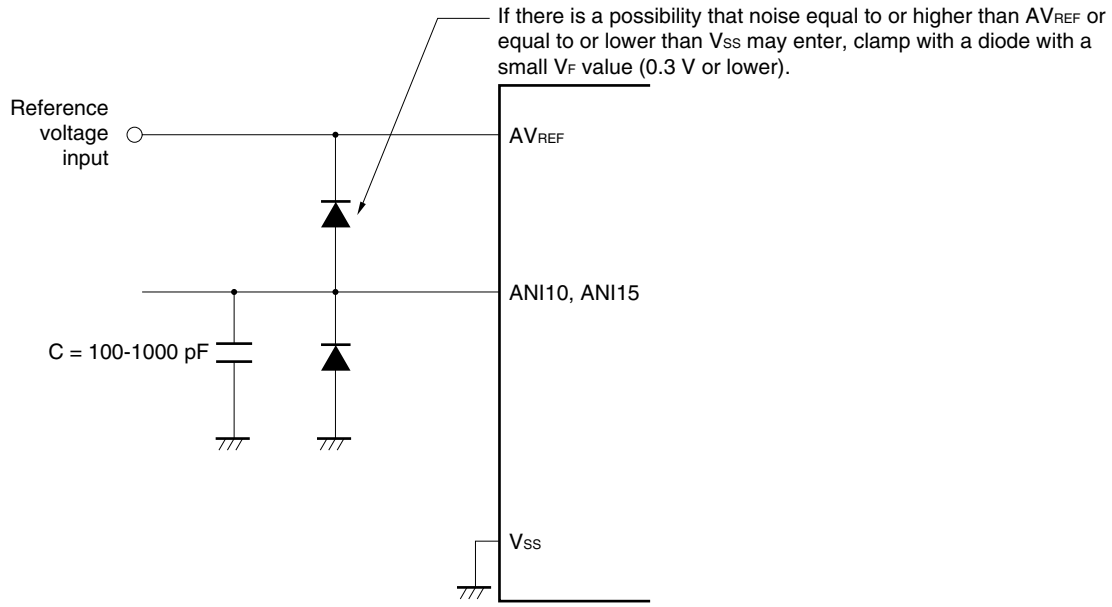
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external C as shown in Figure 10-25 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 10-25. Analog Input Pin Connection



(5) ANI10, ANI15

<1> The analog input pins (ANI10, ANI15) are also used as input port pins (P152, P157).
When A/D conversion is performed with any of ANI10, and ANI15 selected, do not access P152, and P157 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to select pins used as P152, and P157 starting with the ANI10/P152 that is the furthest from AVREF.

<2> If the pins adjacent to the pins currently used for A/D conversion are used as digital I/O port, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, make sure that digital pulses are not input to or output from the pins adjacent to the pin undergoing A/D conversion.

<3> If any pin among pins of port 15 is used as digital output port during A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, make sure that digital pulses are not output to pins of port 15 during A/D conversion.

(6) Input impedance of ANI10, ANI15 pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 1 kΩ, and to connect a capacitor of about 100 pF to the ANI10 and ANI15 pins (see **Figure 10-25**).

(7) AVREF pin input impedance

A series resistor string of several tens of kΩ is connected between the AVREF and VSS pins.
Therefore, if the output impedance of the reference voltage supply is high, this will result in a series connection to the series resistor string between the AVREF and VSS pins, resulting in a large reference voltage error.

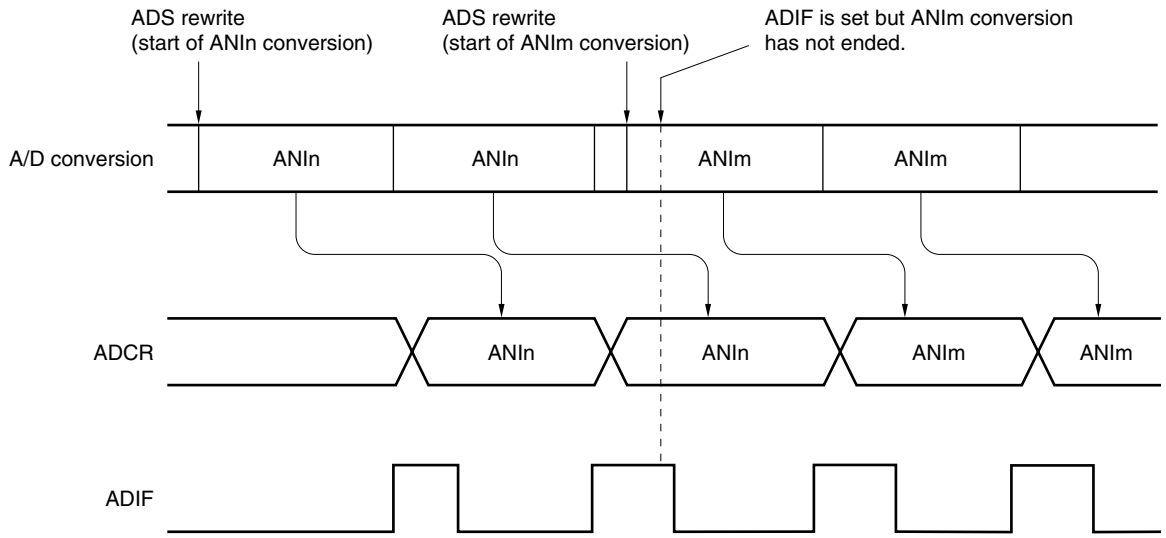
(8) Interrupt request flag (ADIF)

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 for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF 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 before the A/D conversion operation is resumed.

Figure 10-26. Timing of A/D Conversion End Interrupt Request Generation



Remark n = 10, 15,
m = 10, 15

(9) Conversion results just after A/D conversion start

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1, or if the ADCS bit is set to 1 with the ADCE bit = 0. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(10) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register (ADM), A/D converter mode register 1 (ADM1), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR and ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADM1, ADS, or ADPC. Using a timing other than the above may cause an incorrect conversion result to be read.

(11) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 10-27. Internal Equivalent Circuit of ANIn Pin

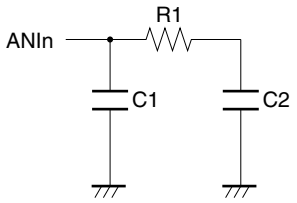


Table 10-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

R1	C1	C2
11.5 kΩ	8.0 pF	8.0 pF

Remarks 1. The resistance and capacitance values shown in Table 10-4 are not guaranteed values.

2. n = 10, 15

CHAPTER 11 24-BIT ΔΣ-TYPE A/D CONVERTER

11.1 Functions of 24-bit ΔΣ-Type A/D Converter

The 24-bit ΔΣ-type A/D converter has a 24-bit resolution when converting an analog input signal to digital values.

- ○ S/N ratio: 62 dB min. (when gain of × 16 is selected for channels 1 and 3)
- ○ 24-bit resolution (conversion result register: 24 bit)
- ○ 4 channels
- ○ Analog input: 8 (positive, negative input/channel)
- ○ ΔΣ conversion mode
- ○ Pre-amplifier gain selectable: × 2 or × 16 (channels 1 and 3)
- ○ Operating voltage: LAV_{DD} = 3.0 to 3.6 V, LAV_{SS} = 0 V
- ○ Analog input voltage: ±0.375 V (channels 0 and 2)
  ±0.1875 V (channels 1 and 3, when pre-amplifier gain of × 2 is selected)
  ±23.4 mV (channels 1 and 3, when pre-amplifier gain of × 16 is selected)
- ○ Reference voltage generation (1.226 V (TYP.) can be output)[Note]
- ○ Conversion rate selectable: 4.340 kHz

Note The reference voltage for the 24-bit ΔΣ-type A/D converter can be selected from the external reference potential and internal reference potential. Table 11-1 shows the reference characteristics.

Table 11-1. Reference Characteristics

Item	Symbol	MIN.	TYP.	MAX.	Unit
external reference potential (input)	AV_{REFIO1}	1.20		1.25	V
internal reference potential (output)	AV_{REFIO2}	1.165	1.226	1.287	V

Caution To use the 24-bit ΔΣ-type A/D converter, supply 10 MHz to the extended SFR (3rd SFR) by setting the clock output select register 0 (CKS0) (see CHAPTER 5 CLOCK GENERATOR)

11.2 Configuration of 24-bit ΔΣ-Type A/D Converter

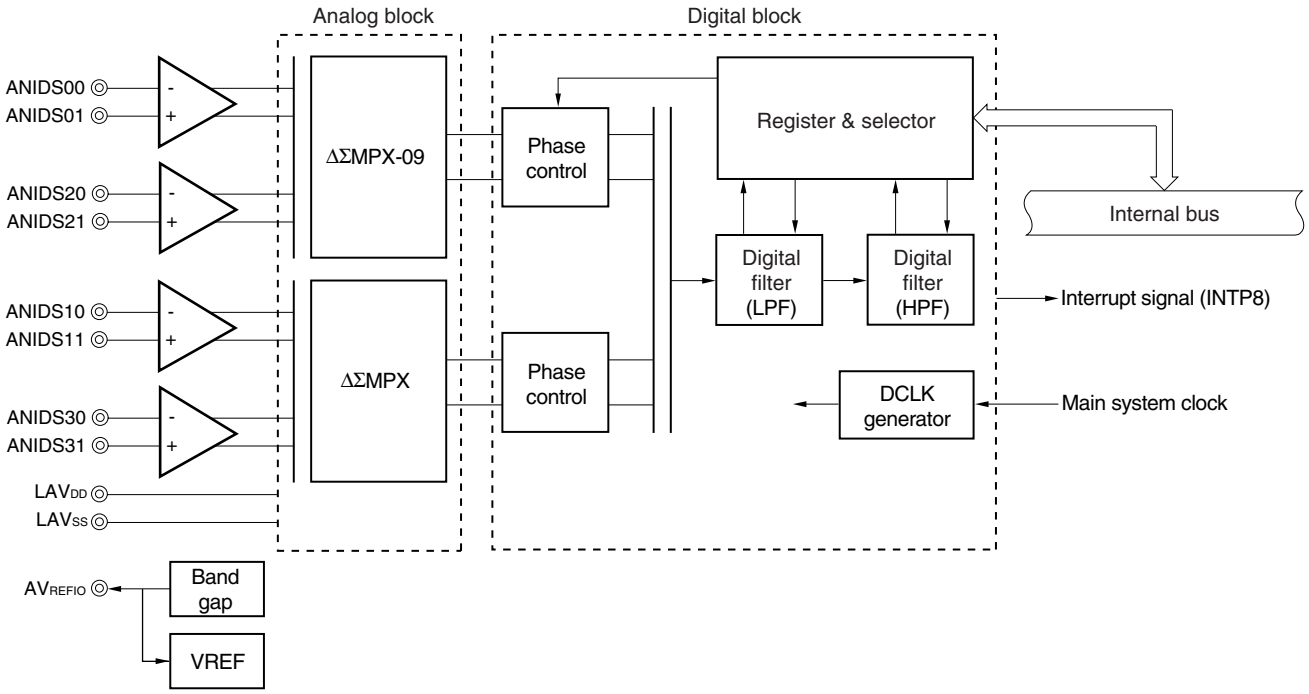
The 24-bit ΔΣ-type A/D converter consists of the following hardware.

Table 11-2. Configuration of 24-bit ΔΣ-Type A/D Converter

Item	Configuration
Analog input	4 channels and 8 inputs
Registers	24-bit ΔΣ-type A/D converter mode register (ADM2) High-pass filter control register 0 (HPFC0) High-pass filter control register 1 (HPFC1) 24-bit ΔΣ-type A/D conversion result register n (ADCRn) Phase control registers 0 and 1 (PHC0 and PHC1) A/D clock delay setting register (ADLY)
Internal units	Pre-amplifier block ΔΣ converter Phase control Reference voltage generator Digital filter (DF) High-pass filter (HPF)

Remark n = 0 to 3

Figure 11-1. Block Diagram of 24-bit ΔΣ-Type A/D Converter



The channels (voltage and current) supported for each input pin differ depending on whether the two-wire mode or three-wire mode is used.

Connect the voltage channel to the voltage sensor and the current channel to the current sensor in accordance with the following tables.

< In the case of two-wire mode (CHMD = 0) >

Input mode	78K0R/LG3-M
ANIDS00, ANIDS01	voltage channel
ANIDS10, ANIDS11	current channel 1
ANIDS20, ANIDS21	current channel 2
ANIDS30, ANIDS31	Not used

< In the case of three-wire mode (CHMD = 1) >

Input mode	78K0R/LG3-M
ANIDS00, ANIDS01	voltage channel 1
ANIDS10, ANIDS11	current channel 1
ANIDS20, ANIDS21	voltage channel 2
ANIDS30, ANIDS31	current channel 2

(1) Pre-amplifier

This unit shifts the signal input to the ANIDSn0 and ANIDSn1 pins with LAV_{SS} as the reference voltage into the internal reference voltage, and then amplifies the input signal. It supplies its output signal to the ΔΣ circuit.

(2) Multiplex ΔΣ circuit

Two 2-multiplex ΔΣ circuits are provided so that a total of 4 channels of analog inputs can be converted into digital signals. These two ΔΣ circuits operate synchronously, and one ΔΣ circuit executes analog input conversion of two channels by time division. The input signal is amplified by the pre-amplifier and ΔΣ circuit, and the gain of channels 0 and 2 are fixed to × 1, and that of channels 1 and 3 can be selected from × 2 and × 16. A high-speed mode (4.340 kHz) is selectable as the conversion rate, and the over-sampling frequency in the respective modes is 555.6 kHz (at operation clock = 10 MHz).

(3) Phase adjustment circuit

This circuit adjusts the phase of input analog signals. The phase between analog signals is adjusted by using a step of 128 fs.

Phase shifts between input analog signals occur due to external components (such as current sensors). Use the PHC register to correct such phase shifts in advance, because these shifts can decrease the precision of power calculations.

Correcting phase shifts uses 255 steps, and each step is equivalent to a 128 fs (1/128 x 4.34 k) cycle, so the delay time is about 1.8 μs. The phase can be adjusted in 0.0389° units if the line frequency is 60 Hz, or in 0.0324° units if the line frequency is 50 Hz.

(4) Reference voltage generator

An internal reference voltage source (band-gap reference circuit) is provided and a reference voltage is output from the reference voltage I/O pin (AV_{REFIO}). To use an external reference voltage source, input its voltage to the AV_{REFIO} pin.

(5) Digital filter (DF)

This unit eliminates high harmonic noise included in the ΔΣ circuit and thins out the data rate to 1/128.

(6) High-pass filter

This unit eliminates the DC component included in the input signal and the DC offset generated by the analog circuit. Whether the high-pass filter is inserted or not can be selected for each channel.

(7) ANIDSn0 to ANIDSn1 pins

These are analog input pins of the A/D converter. One channel inputs two signals. The ANIDSn0 pin is the negative input, while the ANIDSn1 pin is the positive input.

(8) LAV_{DD} pin

This is the analog power supply pin of the A/D converter. Always keep the voltage on this pin the same as that on the LV_{DD} pin even when the A/D converter is not used.

(9) LAV_{SS} pin

This is the ground pin of the A/D converter. Always keep the voltage on this pin the same as that on the LV_{SS} pin even when the A/D converter is not used.

(10) AV_{REFIO} pin

This is the I/O pin of the reference voltage to the A/D converter. This pin is used as an output pin if using an internal reference voltage, or as an input pin if using an external reference voltage.

11.3 Registers Used in 24-bit ΔΣ-Type A/D Converter

The 24-bit ΔΣ-type A/D converter uses the following registers.

These registers are all allocated to the extended SFR (3rd SFR) space.

For details about how to access the extended SFR (3rd SFR) space, see **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE**.

- 24-bit ΔΣ-type A/D converter mode register (ADM2)
- HPF filter control register 0 (HPFC0)
- HPF filter control register 1 (HPFC1)
- 24-bit ΔΣ-type A/D conversion result register n (ADCRn)
- Phase control registers 0 and 1 (PHC0 and PHC1)
- A/D clock delay setting register (ADLY)

Remark n = 0 to 3

(1) 24-bit ΔΣ-Type A/D converter mode register (ADM2)

This register is used to turn the power supply for the 24-bit ΔΣ-type A/D converter on or off, enable or stop conversion operation, switch the single phase mode, and specify the reference potential and amplifier gain.

ADM2 is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation sets this register to 00H.

Figure 11-2. Format of 24-Bit ΔΣ-Type A/D Converter Control Register (ADM2)

Address: A3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM2	ADPON	ADCE2	CHMD	0	BGCUT	PAGS1	0	0

ADPON	Specification of power to 24-bit ΔΣ-type A/D converter
0	Power OFF
1	Power ON

ADCE2	Specification of operation of 24-bit ΔΣ-type A/D converter
0	Stop conversion operation
1	Enable conversion operation

CHMD[Note]	Switch between the single-phase two-wire mode and the single-phase three-wire mode
0	Single-phase two-wire mode
1	Single-phase three-wire mode

BGCUT	Control signal that stops BGR output
0	Specifies the use of an internal reference potential and the use of AV_{REFIO} as an output pin.
1	Stops BGR output and specifies the use of an external reference potential and the use of AV_{REFIO} as an input pin.

PAGS1	Specification of programmable amplifier gain of channels 1 and 3
0	× 2
1	× 16

Note The channels (voltage and current) supported for each input pin differ depending on whether the two-wire mode or three-wire mode is used.
Connect the voltage channel to the voltage sensor and the current channel to the current sensor in accordance with the following tables.

<In the case of two-wire mode (CHMD = 0) >

Input mode	78K0R/LG3-M
ANIDS00, ANIDS01	Voltage channel
ANIDS10, ANIDS11	Current channel 1
ANIDS20, ANIDS21	Current channel 2
ANIDS30, ANIDS31	Not used

<In the case of three-wire mode (CHMD = 1) >

Input mode	78K0R/LG3-M
ANIDS00, ANIDS01	Voltage channel 1
ANIDS10, ANIDS11	Current channel 1
ANIDS20, ANIDS21	Voltage channel 2
ANIDS30, ANIDS31	Current channel 2

Caution Be sure to set bits 0, 1, and 4 to 0.

(2) High-pass filter control register 0 (HPFC0)

This 8-bit register is used to specify insertion of a high-pass filter for each channel.

HPFC0 is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation sets this register to 00H.

Figure 11-3. Format of High-Pass Filter Control Register 0 (HPFC0)

Address: A5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
HPFC0	0	0	0	0	THR3	THR2	THR1	THR0

THRn	Specification of insertion of high-pass filter for channel n (n = 0 to 3)
0	Insert high-pass filter
1	Do not insert high-pass filter and through.

Caution Be sure to set bits 4 to 7 to 0.

(3) High-pass filter control register 1 (HPFC1)

This 8-bit register is used to control the high-pass filter operation for each channel.

HPFC1 is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation sets this register to 00H.

Figure 11-4. Format of High-Pass Filter Control Register 1 (HPFC1)

Address: A6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
HPFC1	0	0	0	0	DLY3	DLY 2	DLY 1	DLY 0

DLY n	Specify phase delay to add to channel n (n = 0 to 3)
0	Does not add phase delay in the high-pass filter.
1	Adds phase delay in the high-pass filter.

Caution Be sure to set bits 4 to 7 to 0.

(4) 24-Bit ΔΣ-Type A/D conversion result registers n (ADCRn)

These 24-bit registers are used to store the conversion result of each channel.

ADCRn is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

The value of these registers is initialized to 0000H by system reset and when the ADCE2 bit is 0.

Figure 11-5. Format of 24-Bit ΔΣ-Type A/D Conversion Result Registers n (ADCRn)

Address: A7H to A9H After reset: 000000H R

Symbol	ADCR0H (A9H)								ADCR0M (A8H)								ADCR0L (A7H)							
ADCR0																								
	23							16	15							8	7							0

Address: AAH to ACH After reset: 000000H R

Symbol	ADCR1H (ACH)								ADCR1M (ABH)								ADCR1L (AAH)							
ADCR1																								
	23							16	15							8	7							0

Address: ADH to AFH After reset: 000000H R

Symbol	ADCR2H (AFH)								ADCR2M (AEH)								ADCR2L (ADH)							
ADCR2																								
	23							16	15							8	7							0

Address: B0H to B2H After reset: 000000H R

Symbol	ADCR3H (B2H)								ADCR3M (B1H)								ADCR3L (B0H)							
ADCR3																								
	23							16	15							8	7							0

Cautions 1. Read the ADCRn register when the ADCE2 bit is 1, because the register is initialized when the ADCE2 bit is 0.

2. For full-scale input, the ADCRn register is set to the value below.

Input voltage (gain 1 time)	ADCRn register value
0.375 V	0x400000
–0.375 V	0xC00000

Remark n = 0 to 3

(5) Phase control registers 0 and 1 (PHC0, PHC1)

These 9-bit registers are used to control the phase adjustment for each channel.

Phase adjustment changes the timing at which one bit of A/D conversion data input from the analog block is output to the digital filter block. These registers can be used to perform 256 steps of adjustment.

PHC0 and PHC1 are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation sets this register to 0000H.

Figure 11-6. Format of Phase Control Registers 0 and 1 (PHC0, PHC1)

Address: B3H and B4H After reset: 0000H R/W

Symbol	PHC0H (B4H)							PHC0L (B3H)								
PHC0	0	0	0	0	0	0	0	PHC08	PHC07	PHC06	PHC05	PHC04	PHC03	PHC02	PHC01	PHC00
	15							8	7							0

Address: B5H and B6H After reset: 0000H R/W

Symbol	PHC1H (B6H)							PHC1L (B5H)								
PHC1	0	0	0	0	0	0	0	PHC18	PHC17	PHC16	PHC15	PHC14	PHC13	PHC12	PHC11	PHC10
	15							8	7							0

PHC0L (PHC00 to PHC07), PHC1L (PHC10 to PHC17)	Specify phase adjustment amount (one step = 128 fs)
00H	Through (no phase adjustment)
01H	One step
…	
FEH	254 steps
FFH	255 steps

PHC08, PHC18	Select voltage channel or current channel for phase adjustment
0	Specifies that the value specified for PHCnL be used to adjust the phase of voltage channel n (n = 0 or 1).
1	Specifies that the value specified for PHCnL be used to adjust the phase of current channel n (n = 0 or 1).

(6) A/D clock delay setting register (ADLY)

ADLY is a register that controls the phase between the A/D operation clock and the digital clock.

Be sure to clear this register to 00H.

ADLY is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation sets this register to 00H.

Figure 11-7. Format of A/D Clock Delay Setting Register (ADLY)

Address: A4H After reset: 00H R/W

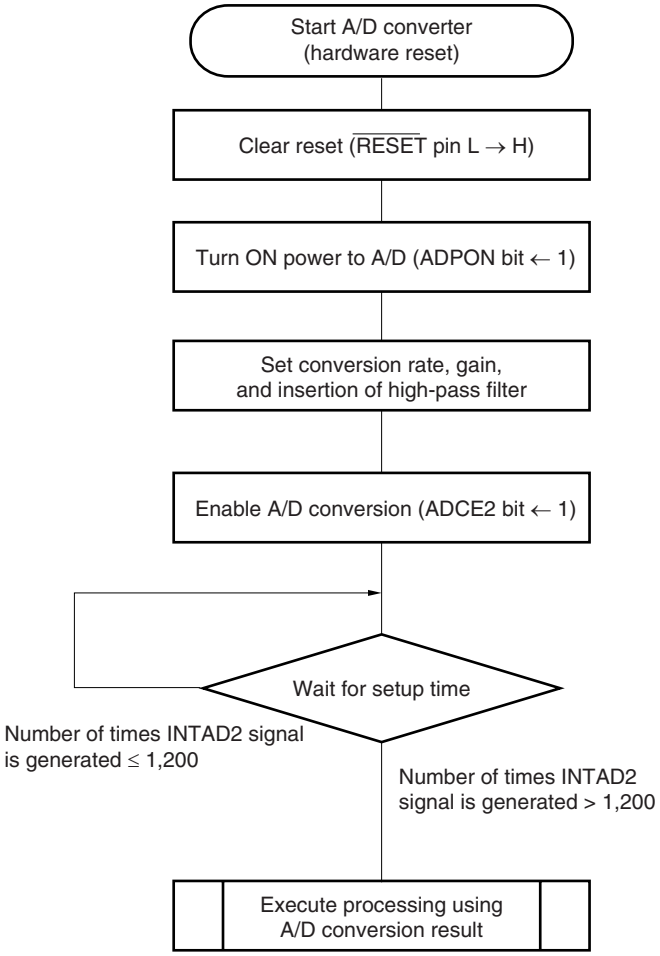
Symbol	7	6	5	4	3	2	1	0
ADLY	0	0	0	0	0	0	0	0

Caution Be sure to set bits 0 to 7 to 0.

11.4 Operation of 24-bit ΔΣ-Type A/D Converter

The A/D converter starts operating when the ADPON bit and ADCE2 bit of ADM2 register are set to 1. The setup time of the analog block and digital filter block is required after power application and start of conversion. Perform initialization in accordance with the flowchart below.

Figure 11-8. Initialization Flowchart



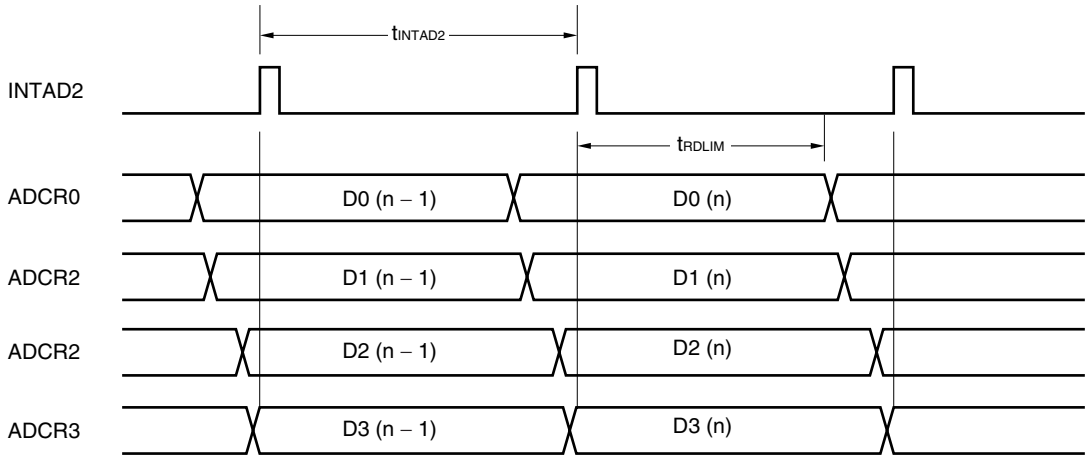
Caution If the A/D converter is temporarily stopped for initialization (ADCE2 bit ← 0 with ADPON bit set to 1) and then restarted, it is necessary to wait for a certain setup time. In this case, the setup time should be equal to 10 A/D conversion end interrupt request signals (INTAD2), which is the delay of the digital filter.

When A/D conversion is enabled, conversion of the signals on the four channels of analog input pins (ANIDSn0 and ANIDSn1 pins) is started. Two sets of 2-multiplex ΔΣ circuits are provided, each of which executes conversion of two channels by time division. Each time conversion of all the four channels is completed, the INTAD2 signal is generated to inform the CPU that the conversion result can be read.

To read the ADCRn register by interrupt servicing, the maximum pending time is as shown in **Figure 11-9**. Complete reading of the ADCRn register within this time.

Remark n = 0 to 3

Figure 11-9. Timing of Generation of INTAD2 Signal and Storing in ADCRn Register (operation clock = 10 MHz)



t_{INTAD2}: Interrupt generation cycle : 230.4 μs

t_{RDLIM}: ADCRn register read pending time (MAX.): 223.8 μs (t_{INTAD2} – 5.5 × A/D system clock (operation clock/12))

11.5 Cautions of 24-bit ΔΣ-Type A/D Converter

(1) Read the ADCRn register by A/D conversion end interrupt (INTAD2) servicing. Otherwise, an illegal value may be read because of a conflict between storing the conversion value in the ADCRn register and reading the register. The period of the INTAD2 processing during which reading the ADCRn register is 223.8 μs (at 10 MHz).

(2) After turning ON power to the A/D converter (ADPON bit of ADM2 register is set to 1), the internal setup time of the A/D converter is necessary. Consequently, the data of the first 1,200 conversions is invalid.

<R> **Caution Count the INTAD2 signal 1,200 times after the A/D converter operation is started and then load the converted data when the next INTAD2 signal is generated.**

(3) The setup time is also necessary when the A/D converter has been temporarily stopped once for initialization (by clearing the ADCE2 bit of ADM2 register with the ADPON bit set to 1) and then restarted. Wait for the duration of 10 INTAD2 signals, which is the delay of the digital filter.

(4) The time required for the correct data to be output after the conversion operation has been enabled (by setting the ADCE2 bit to 1) differs depending on the analog input status at that time. This is because the stabilization time of the digital filter changes depending on the analog input status. Thoroughly evaluate the stabilization time in the environment in which the A/D converter is used.

(5) When the supply clock to the 24-bit ΔΣ-Type A/D converter stops, be sure to initialize the A/D converter. To initialize the A/D converter, first clear (set to 0) and then set (set to 1) the ADCE2 bit. Wait for the duration of 10 INTAD2 signals, which is the delay of the digital filter.

(6) Before turning off LAV_{DD}, be sure to clear ADPON bit and ADCE2 bit of ADM2 register to 0.

(7) Be sure to set the ADM2 register while the A/D converter is stopped (ADCE2 bit = 0).

(8) Because the ADCRn register is initialized when the ADCE2 bit is 0, read the ADCRn register when the ADCE2 bit is 1.

(9) Clear the ADPON bit to 0 before shifting to the software STOP mode.
If software STOP mode is entered with the ADPON bit set to 1, a current will flow.

(10) When reading data from the ADCRn register, read the lower bytes of data.

(11) For the digital filter, a gain of approximately 0.962 is added.

(12) To use the 24-bit ΔΣ-type A/D converter, supply 10 MHz to the extended SFR (3rd SFR) by setting the clock output selection register (CKS) (see **CHAPTER 5 CLOCK GENERATOR**).

Remark n = 0 to 3

CHAPTER 12 SERIAL ARRAY UNIT

The serial array unit has four serial channels per unit and can use two or more of various serial interfaces (3-wire serial (CSI), UART, and simplified I^2C) in combination.

Function assignment of each channel supported by the 78K0R/Lx3-M microcontrollers is as shown below. Channel 2 of unit 0 is dedicated to the extended SFR (3rd SFR) interface and channels 2 and 3 of unit 1 are dedicated to UART3 (supporting LIN-bus).

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	–	UART0	–
	1	–		–
	2	CSI10 (dedicated to the extended SFR (3rd SFR) interface)	–	–
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

(Example of combination) When “UART2” is used for channels 0 and 1 of unit 1, CSI20 and IIC20 cannot be used, but UART3 (supporting LIN-bus) can be used.

12.1 Functions of Serial Array Unit

Each serial interface supported by the 78K0R/Lx3-M microcontrollers has the following features.

12.1.1 3-wire serial I/O (CSI20)

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
- Level setting of transmit/receive data

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

12.1.2 UART (UART0, UART2, UART3)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. It transmits or receives data in asynchronization with the party of communication (by using an internal baud rate). Full-duplex UART communication can be realized by using two channels, one dedicated to transmission (even channel) and the other to reception (odd channel).

[Data transmission/reception]

- Data length of 5, 7, or 8 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

The LIN-bus is accepted in UART3 (2 and 3 channels of unit 1)

[LIN-bus functions]

- Wakeup signal detection
- Sync break field (SBF) detection
- Sync field measurement, baud rate calculation

} External interrupt (INTP0) or timer array unit (TAU) is used.

12.1.3 Simplified I²C (IIC20)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master and does not have a function to detect wait states.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[Note] and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Parity error (ACK error)

[Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

Note An ACK is not output when the last data is being received by writing 0 to the SOE10 (SOE1 register) bit and stopping the output of serial communication data. See **12.7.3 (2) Processing flow** for details.

Remark To use an I²C bus of full function, see **CHAPTER 13 SERIAL INTERFACE IICA**.

12.2 Configuration of Serial Array Unit

Serial array unit includes the following hardware.

Table 12-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits
Buffer register	Lower 8 bits of serial data register mn (SDRmn)[Note]
Serial clock I/O	$\overline{\text{SCK20}}$ pin (for 3-wire serial I/O), SCL20 pin (for simplified I^2C)
Serial data input	SI20 pin (for 3-wire serial I/O), RxD0, RxD2 pins (for UART), RxD3 pin (for UART supporting LIN-bus)
Serial data output	SO20 pin (for 3-wire serial I/O), TxD0, TxD2 pins (for UART), TxD3 pin (for UART supporting LIN-bus), output controller
Serial data I/O	SDA20 pin (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn) • Port input mode registers 1 (PIM1) • Port output mode registers 1, 8 (POM1, POM8) • Port mode registers 1, 5, 8 (PM1, PM5, PM8) • Port registers 1, 5, 8 (P1, P5, P8)

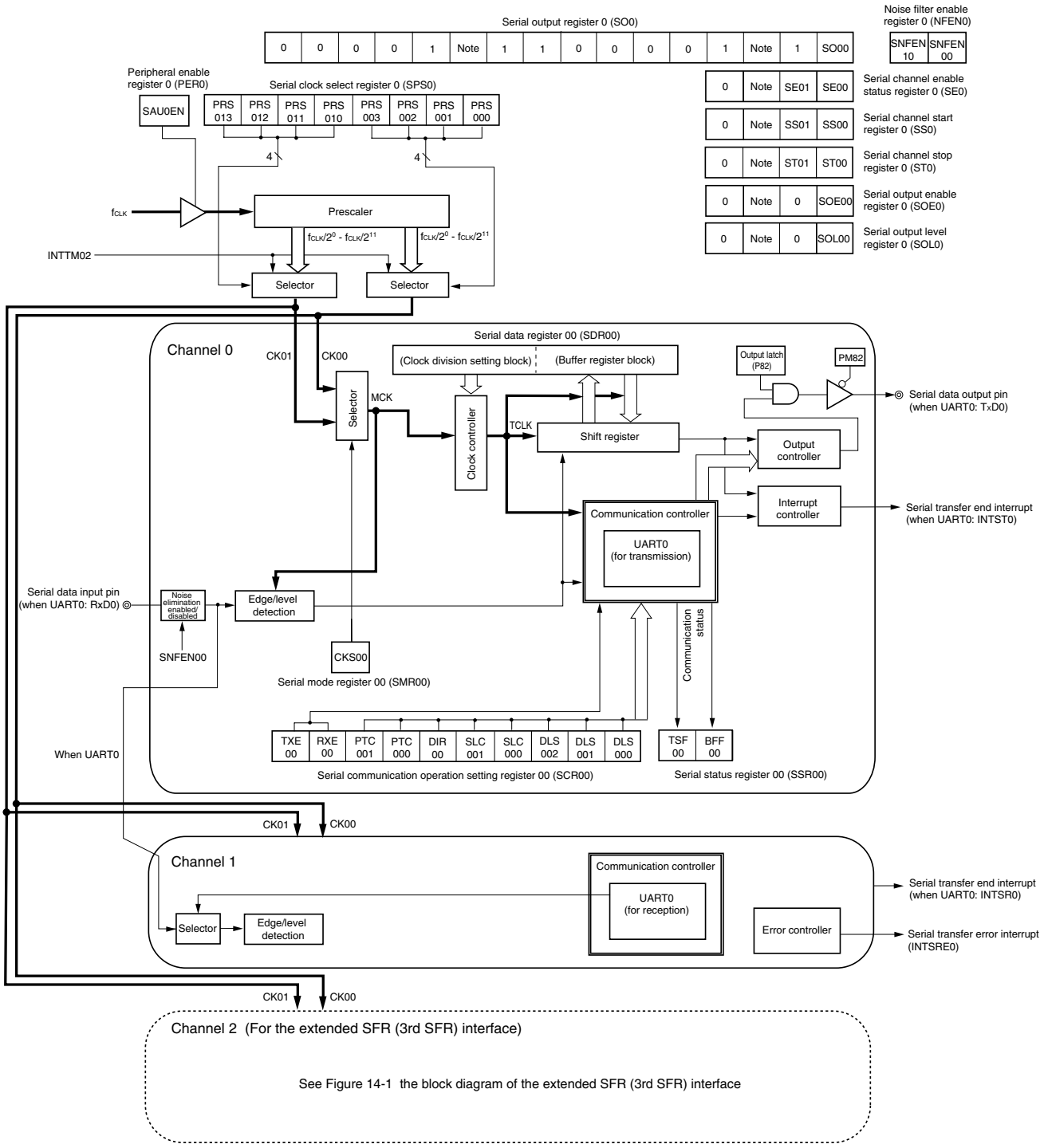
Note The lower 8 bits of the serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

- 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 = 20),
q: UART number (q = 0, 2, 3), r: IIC number (r = 20)

Figure 12-1 shows the block diagram of serial array unit 0.

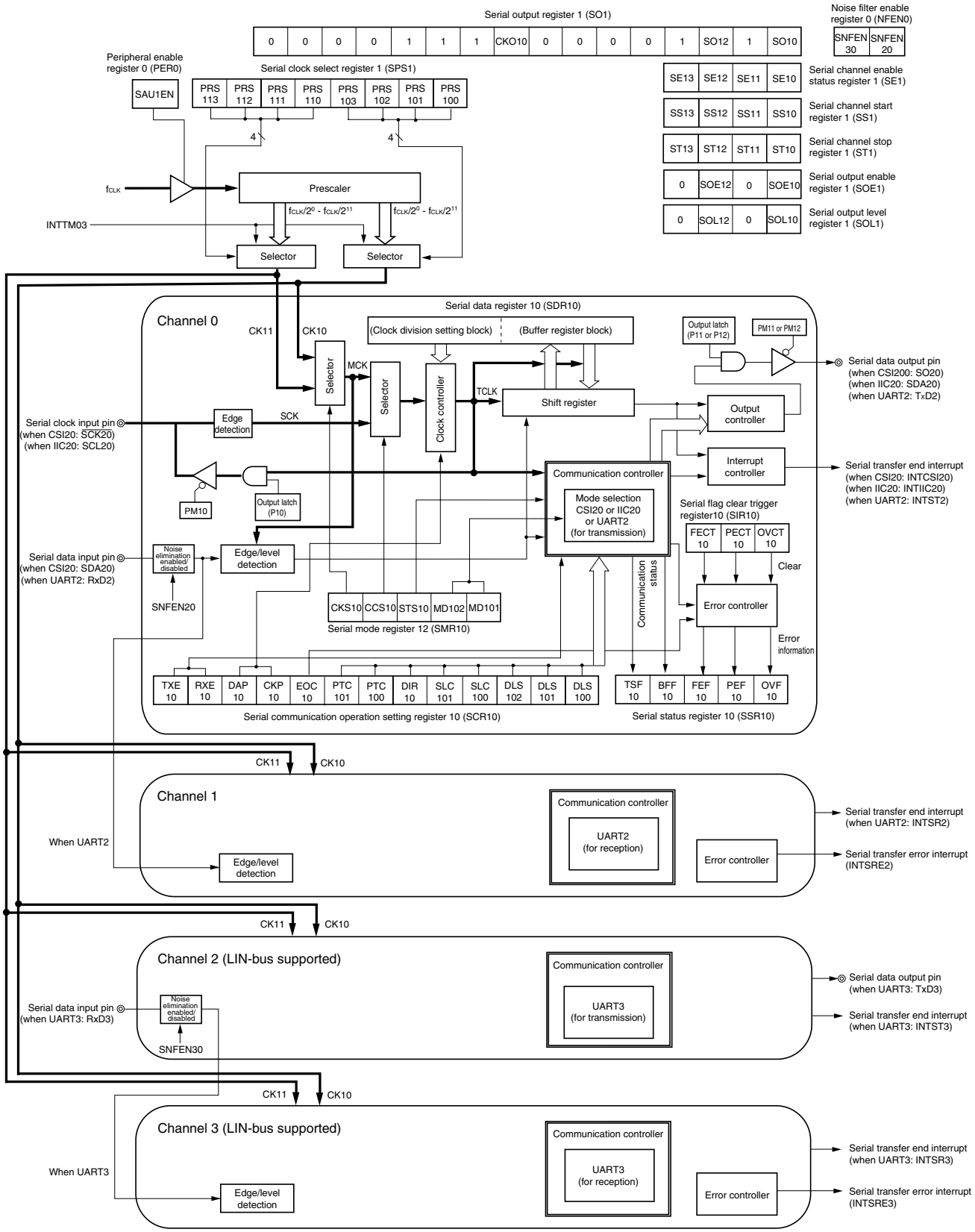
Figure 12-1. Block Diagram of Serial Array Unit 0



Remark These are bits used for extended SFR (3rd SFR).
See **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE**.

Figure 12-2 shows the block diagram of serial array unit 1.

Figure 12-2. Block Diagram of Serial Array Unit 1



(1) Shift register

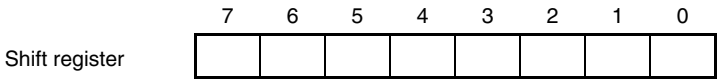
This is an 8-bit register that converts parallel data into serial data or vice versa.

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 8 bits of serial data register mn (SDRmn).



(2) Lower 8 bits of the serial data register mn (SDRmn)

SDRmn is the transmit/receive data register (16 bits) of channel n. Bits 7 to 0 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 (MCK).

When data is received, parallel data converted by the shift register is stored in the lower 8 bits. When data is to be transmitted, set transmit to be transferred to the shift register to the lower 8 bits.

The data stored in the lower 8 bits of this register is as follows, depending on the setting of bits 0 to 2 (DLSmn0 to DLSmn2) of the SCRmn register, regardless of the output sequence of the data.

- 5-bit data length (stored in bits 0 to 4 of SDRmn register) (settable in UART mode only)
- 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)

SDRmn can be read or written in 16-bit units.

The lower 8 bits of SDRmn of SDRmn can be read or written[Note] 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 this register to 0000H.

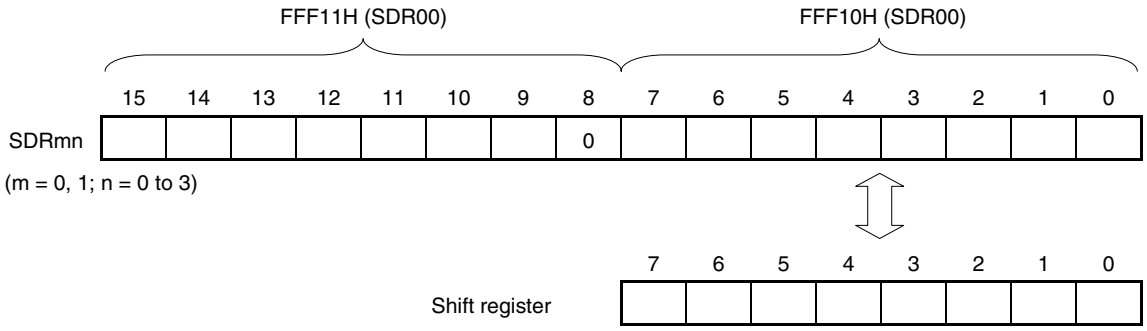
Note Writing in 8-bit units is prohibited when the operation is stopped (when SEmn = 0).

Remarks 1. After data is received, “0” is stored in bits 0 to 7 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 = 20), q: UART number (q = 0, 2, 3), r: IIC number (r = 20)

Figure 12-3. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11), FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13) After reset: 0000H R/W



Caution Be sure to clear bit 8 to "0".

Remarks 1. For the function of the higher 7 bits of SDRmn, see **12.3 Registers Controlling Serial Array Unit**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

12.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial status register mn (SSRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial channel enable status register m (SEm)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 1 (PIM1)
- Port output mode registers 1, 8 (POM1, POM8)
- Port mode registers 1, 5, 8 (PM1, PM5, PM8)
- Port registers 1, 5, 8 (P1, P5, P8)

Remark m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 3)

(1) Peripheral enable register 0 (PER0)

PER0 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 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.

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

Reset signal generation clears this register to 00H.

Figure 12-4. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

SAUmEN	Control of serial array unit m input clock
0	Stops supply of input clock. • SFR used by serial array unit m cannot be written. • Serial array unit m is in the reset status.
1	Supplies input clock. • SFR used by serial array unit m can be read/written.

Cautions 1. When setting serial array unit m, be sure to set SAUmEN to 1 first. 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 (except for input switch control register (ISC), noise filter enable register (NFEN0), port input mode register (PIM1), port output mode registers (POM1, POM8), port mode registers (PM1, PM5, PM8), and port registers (P1, P5, P8)).

2. After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1)

(2) Serial clock select register m (SPSm)

SPSm 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 SPSm, and CKm0 is selected by bits 3 to 0.

Rewriting SPSm is prohibited when the register is in operation (when SEmn = 1).

SPSm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SPSm can be set with an 8-bit memory manipulation instruction with SPSmL.

Reset signal generation clears this register to 0000H.

Figure 12-5. 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 mp3	PRS mp2	PRS mp1	PRS mp0	Section of operation clock (CKmp)[Note 1]				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	1	1	INTTM02 if m = 0, INTTM03 if m = 1[Note 2]				
Other than above				Setting prohibited				

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (STm = 000FH) the operation of the serial array unit (SAUm). When selecting INTTM02 and INTTM03 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAUm can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting TIS02 (if m = 0) or TIS03 (if m = 1) of the TIS0 register to 1) and selecting INTTM02 and INTTM03 by using the SPSm register in channels 2 and 3 of TAU0. When changing f_{CLK}, however, SAUm and TAU0 must be stopped as described in Note 1 above.

Cautions 1. Be sure to clear bits 15 to 8 to "0".

2. After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

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

2. m: Unit number (m = 0, 1), p = 0, 1

(3) Serial mode register mn (SMRmn)

SMRmn is a register that sets an operation mode of channel n. It is also used to select an operation clock (MCK), specify whether the serial clock (SCK) may be input or not, set a start trigger, an operation mode (CSI, UART, or I^2C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting SMRmn is prohibited when the register is in operation (when SEmn = 1). However, the MDmn0 bit can be rewritten during operation.

SMRmn can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets this register to 0020H.

Figure 12-6. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00), F0112H, F0113H (SMR01), F0150H, F0151H (SMR10), F0152H, F0153H (SMR11), F0154H, F0155H (SMR12), F0156H, F0157H (SMR13) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn	0	SISmn0	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operation clock (MCK) of channel n
0	Prescaler output clock CKm0 set by SPSm register
1	Prescaler output clock CKm1 set by SPSm register
Operation clock MCK is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (TCLK) is generated.	

CCSmn	Selection of transfer clock (TCLK) of channel n
0	Divided operation clock MCK specified by CKSmn bit
1	Clock input from SCK pin (slave transfer in CSI mode)
Transfer clock TCLK is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of MCK is set by the higher 7 bits of the SDRmn register.	

STSmn	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I^2C).
1	Valid edge of RxD pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13.

Figure 12-6. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00), F0112H, F0113H (SMR01), F0150H, F0151H (SMR10), F0152H, F0153H (SMR11), F0154H, F0155H (SMR12), F0156H, F0157H (SMR13) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn	0	SIS mn0	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^2C mode
1	1	Setting prohibited

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt
For successive transmission, the next transmit data is written by setting MDmn0 to 1 when SDRmn data has run out.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3) mn = 00, 01, 10 to 13.

(4) Serial communication operation setting register mn (SCRmn)

SCRmn 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 SCRmn is prohibited when the register is in operation (when SEmn = 1).

SCRmn can be set by a 16-bit memory manipulation instruction.

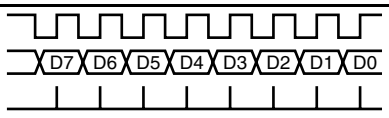
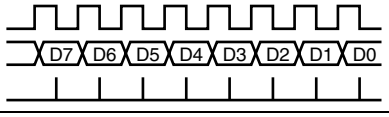
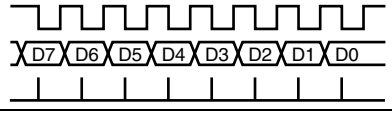
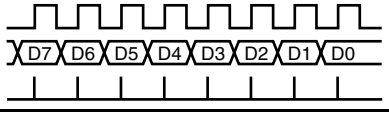
Reset signal generation sets this register to 0087H.

Figure 12-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00), F011AH, F011BH (SCR01), F0158H, F0159H (SCR10), F015AH, F015BH (SCR11), F015CH, F015DH (SCR12), F015EH, F015FH (SCR13) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Does not start 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		1
0	1		2
1	0		3
1	1		4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C mode.			

EOC mn	Selection of masking of error interrupt signal (INTSREx (x = 0, 2, 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^2C mode, and during UART transmission. Set EOCmn = 1 during UART reception.	

Caution Be sure to clear bits 3, 6, and 11 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 = 20), mn = 00, 01, 10 to 13.

Figure 12-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00), F011AH, F011BH (SCR01), F0158H, F0159H (SCR10), F015AH, F015BH (SCR11), F015CH, F015DH (SCR12), F015EH, F015FH (SCR13) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

PTC mn1	PTC mn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity.	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the CSI mode and simplified I^2C 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^2C mode.	

SLC mn1	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
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^2C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the CSI mode.		

DLS mn2	DLS mn1	DLS mn0	Setting of data length in CSI and UART modes
1	0	0	5-bit data length (stored in bits 0 to 4 of SDRmn register) (settable in UART mode only)
1	1	0	7-bit data length (stored in bits 0 to 6 of SDRmn register)
1	1	1	8-bit data length (stored in bits 0 to 7 of SDRmn register)
Other than above			Setting prohibited
Be sure to set DLSmn0 = 1 in the simplified I^2C mode.			

Caution Be sure to clear bits 3, 6, and 11 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 = 20), mn = 00, 01, 10 to 13.

(5) Higher 7 bits of the serial data register mn (SDRmn)

SDRmn is the transmit/receive data register (16 bits) of channel n. Bits 7 to 0 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 (MCK).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by the higher 7 bits of SDRmn is used as the transfer clock.

For the function of the lower 8 bits of SDRmn, see **12.2 Configuration of Serial Array Unit**.

SDRmn can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8 bits of SDRmn. When SDRmn is read during operation, 0 is always read.

Reset signal generation clears this register to 0000H.

Figure 12-8. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11), FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13) After reset: 0000H R/W

FFF11H (SDR00) FFF10H (SDR00)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0								

SDRmn[15:9]							Transfer clock setting by dividing the operating clock (MCK)
0	0	0	0	0	0	0	MCK/2
0	0	0	0	0	0	1	MCK/4
0	0	0	0	0	1	0	MCK/6
0	0	0	0	0	1	1	MCK/8
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	MCK/254
1	1	1	1	1	1	1	MCK/256

Cautions 1. Be sure to clear bit 8 to "0".

2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

3. Setting SDRmn[15:9] = 0000000B is prohibited when the simplified I^2C is used. Set SDRmn[15:9] to 0000001B or greater.

Remarks 1. For the function of the lower 8 bits of SDRmn, see **12.2 Configuration of Serial Array Unit**.

2. m: Unit number (m = 0, 1) n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13.

(6) Serial status register mn (SSRmn)

SSRmn 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.

SSRmn can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of SSRmn can be set with an 8-bit memory manipulation instruction with SSRmnL.

Reset signal generation clears this register to 0000H.

Figure 12-9. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01), F0140H, F0141H (SSR10), F0142H, F0143H (SSR11), F0144H, F0145H (SSR12), F0146H, F0147H (SSR13) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn Note	PEF mn Note	OVF mn Note

TSF mn	Communication status indication flag of channel n
0	Communication is not under execution.
1	Communication is under execution.
Because this flag is an updating flag, it is automatically cleared when the communication operation is completed. This flag is cleared also when the STmn/SSmn bit is set to 1.	

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.
This is an updating flag. It is automatically cleared when transfer from the SDRmn register to the shift register is completed. During reception, it is automatically cleared when data has been read from the SDRmn register. This flag is cleared also when the STmn/SSmn bit is set to 1. This flag is automatically set if transmit data is written to the SDRmn register when the TXEmn bit of the SCRmn register = 1 (transmission or reception mode in each communication mode). It is automatically set if receive data is stored in the SDRmn register when the RXEmn bit of the SCRmn register = 1 (transmission or reception mode in each communication mode). It is also set in case of a reception error. 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 (OVFmn = 1) is detected.	

Note Only SSR00 and SSR12 registers do not have FET00, FET12, PET00, PET12, OVF00, and OVF12.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13.

Figure 12-9. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01), F0140H, F0141H (SSR10), F0142H, F0143H (SSR11), F0144H, F0145H (SSR12), F0146H, F0147H (SSR13) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn Note	PEF mn Note	OVF mn Note

FEF mn	Framing error detection flag of channel n
0	No error occurs.
1	A framing error occurs during UART reception. <Framing error cause> A framing error occurs if the stop bit is not detected upon completion of UART reception.
This is a cumulative flag and is not cleared until 1 is written to the FECTmn bit of the SIRmn register.	

PEF mn	Parity error detection flag of channel n
0	Error does not occur.
1	A parity error occurs during UART reception or ACK is not detected during I^2C transmission. <Parity error cause> • A parity error occurs if the parity of transmit data does not match the parity bit on completion of UART reception. • ACK is not detected if the ACK signal is not returned from the slave in the timing of ACK reception during I^2C transmission.
This is a cumulative flag and is not cleared until 1 is written to the PECTmn bit of the SIRmn register.	

OVF mn	Overrun error detection flag of channel n
0	No error occurs.
1	An overrun error occurs. <Causes of overrun error> • Receive data stored in the SDRmn register is not read and transmit data is written or the next receive data is written. • Transmit data is not ready for slave transmission or reception in the CSI mode.
This is a cumulative flag and is not cleared until 1 is written to the OVCTmn bit of the SIRmn register.	

Note Only SSR00 and SSR12 register do not have FET00, FET12, PET00, PET12, OVF00, and OVF12.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13.

(7) Serial flag clear trigger register mn (SIRmn)

SIRmn 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 SIRmn is a trigger register, it is cleared immediately when the corresponding bit of SSRmn is cleared.

SIRmn can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SIRmn can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears this register to 0000H.

Figure 12-10. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F010AH, F010BH (SIR01), F0148H, F0149H (SIR10), F014AH, F014BH (SIR11), F014EH, F014FH (SIR13) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FEC Tmn	PEC Tmn	OVC Tmn

FEC Tmn	Clear trigger of framing error of channel n
0	No trigger operation
1	Clears the FEFmn bit of the SSRmn register to 0.

PEC Tmn	Clear trigger of parity error flag of channel n
0	No trigger operation
1	Clears the PEFmn bit of the SSRmn register to 0.

OVC Tmn	Clear trigger of overrun error flag of channel n
0	No trigger operation
1	Clears the OVFmn bit of the SSRmn register to 0.

Caution Be sure to clear bits 15 to 3 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 01, 10, 11, 13

2. When the SIRmn register is read, 0000H is always read.

(8) Serial channel enable status register m (SEm)

SEm indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written a bit of serial channel start register 0 (SSm), the corresponding bit of this register is set to 1. When 1 is written a bit of serial channel stop register 0 (STm), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of CKOmn of the 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 CKOmn 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.

SEm can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of SEm can be set with an 1-bit or 8-bit memory manipulation instruction with SEmL.

Reset signal generation clears this register to 0000H.

Figure 12-11. Format of Serial Channel Enable Status Register m (SEm)

Address: F0120H, F0121H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	SE01	SE00

Address: F0160H, F0161H 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

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops (stops with the values of the control register and shift register, and the statuses of the serial clock I/O pin, serial data output pin, and the FEF, PEF, and OVF error flags retained[Note]).
1	Operation is enabled.

Note Bits 6 and 5 (TSFmn, BFFmn) of the SSRmn register are cleared.

Caution Be sure to clear bits 15 to 3 of SE0 and bits 15 to 4 of SE1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13.
×: Bit can be used in extended SFR (3rd SFR) interface only.
See **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE.**

(9) Serial channel start register m (SSm)

SSm 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. Because SSmn is a trigger bit, it is cleared immediately when SEmn = 1.

SSm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SSm can be set with an 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears this register to 0000H.

Figure 12-12. Format of Serial Channel Start Register m (SSm)

Address: F0122H, F0123H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	SS01	SS00

Address: F0162H, F0163H 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 SEmn to 1 and enters the communication wait status (if a communication operation is already under execution, the operation is stopped and the start condition is awaited).

Caution Be sure to clear bits 15 to 3 of SS0 and bits 15 to 4 of SS1 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13.

×: Bit can be used in extended SFR (3rd SFR) interface only.

See **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE.**

2. When the SSm register is read, 0000H is always read.

(10) Serial channel stop register m (STm)

STm 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. Because STmn is a trigger bit, it is cleared immediately when SEmn = 0.

STm can set written by a 16-bit memory manipulation instruction.

The lower 8 bits of STm can be set with an 1-bit or 8-bit memory manipulation instruction with STmL.

Reset signal generation clears this register to 0000H.

Figure 12-13. Format of Serial Channel Stop Register m (STm)

Address: F0124H, F0125H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	ST01	ST00

Address: F0164H, F0165H After reset: 0000H R/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

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears SEmn to 0 and stops the communication operation. (Stops with the values of the control register and shift register, and the statuses of the serial clock I/O pin, serial data output pin, and the FEF, PEF, and OVF error flags retained[Note].)

Note Bits 6 and 5 (TSFmn, BFFmn) of the SSRmn register are cleared.

Caution Be sure to clear bits 15 to 3 of ST0 and bits 15 to 4 of ST1 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13.

×: Bit can be used in extended SFR (3rd SFR) interface only.

See **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE.**

2. When the STm register is read, 0000H is always read.

(11) Serial output enable register m (SOEm)

SOEm 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 SOmn of the 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 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.

SOEm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SOEm can be set with an 1-bit or 8-bit memory manipulation instruction with SOEmL.

Reset signal generation clears this register to 0000H.

Figure 12-14. Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	0	SOE00

Address: F016AH, F016BH 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	SOE12	0	SOE10

SOEmn	Serial output enable/disable 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 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00, 10, 12

×: Bit can be used in extended SFR (3rd SFR) interface only.

See **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE.**

(12) Serial output register m (SOm)

SOm is a buffer register for serial output of each channel.

The value of bit n of this register is output from the serial data output pin of channel n.

The value of bit (n + 8) of this register is output from the serial clock output pin of channel n.

SOmn 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.

CKOmn 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 CKOmn can be changed only by a serial communication operation.

To use the P10/$\overline{\text{SCK20}}$/SCL20, P11/SI20/SDA20/RxD2/INTP6, P12/SO20/TxD2/TO02, P51/TxD3/SEG38, or P82/TxD0 pin as a port function pin, set the corresponding CKOmn and SOmn bits to "1".

SOm can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0F0FH.

Figure 12-15. Format of Serial Output Register m (SOm)

Address: F0128H, F0129H 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	×	1	1	0	0	0	0	1	×	1	SO00

Address: F0168H, F0169H 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	CKO10	0	0	0	0	1	SO12	1	SO10

CKOmn	Serial clock output of channel n
0	Serial clock output value is "0".
1	Serial clock output value is "1".

SOmn	Serial data output of channel n
0	Serial data output value is "0".
1	Serial data output value is "1".

Caution Be sure to set bits 11, 9, 8, 3, and 1 of SO0, and bits 11 to 9, 3, and 1 of SO1 to "1". And be sure to clear bits 15 to 12 and 7 to 4 of SOm to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00, 10, 12

×: Bit can be used in extended SFR (3rd SFR) interface only.
See **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE.**

(13) Serial output level register m (SOLm)

SOLm 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 0000H in the CSI mode and simplifies I^2C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting SOLm is prohibited when the register is in operation (when SEmn = 1).

SOLm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SOLm can be set with an 8-bit memory manipulation instruction with SOLmL.

Reset signal generation clears this register to 0000H.

Figure 12-16. Format of Serial Output Level Register m (SOLm)

Address: F0134H, F0135H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL00

Address: F0174H, F0175H 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	SOL12	0	SOL10

SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution Be sure to clear bits 15 to 1 of SOL0, and bits 15 to 3, and 1 of SOL1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 10, 12

(14) Input switch control register (ISC)

ISC is used to realize a LIN-bus communication operation by UART3 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 (RxD3) 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 (RxD3) pin is selected as a timer input, so that the pulse widths of a sync break field and a sync field can be measured by the timer.

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

Reset signal generation clears this register to 00H.

Figure 12-17. Format of Input Switch Control Register (ISC)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	ISC4	ISC3	ISC2	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 RxD3 pin is used as timer input (wakeup signal detection).

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 RxD3 pin as an external interrupt (to measure the pulse widths of the sync break field and sync field).

Caution Be sure to clear bits 7 to 5 to "0".

Remark Bits 2 to 4 of ISC are not used with SAU1.

(15) Noise filter enable register 0 (NFEN0)

NFEN0 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^2C communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is enabled, CPU/peripheral operating clock (f_{CLK}) is synchronized with 2-clock match detection.

NFEN0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12-18. Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0060H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	SNFEN30	0	SNFEN20	0	0	0	SNFEN00

SNFEN30	Use of noise filter of RxD3/P50/SEG39 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN30 to 1 to use the RxD3 pin. Clear SNFEN30 to 0 to use the P50 or SEG39 pins.	

SNFEN20	Use of noise filter of RxD2/P11/SI20/SDA20/INTP6 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN20 to 1 to use the RxD2 pin. Clear SNFEN20 to 0 to use the P11, SI20, SDA20 or INTP6 pins.	

SNFEN00	Use of noise filter of RxD0/P81/INTP9 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN00 to 1 to use the RxD0 pin. Clear SNFEN00 to 0 to use the P81 or INTP9.	

Caution Be sure to clear bits 5, 3 to 1 to "0".

(16) Port input mode registers 1 (PIM1)

These registers set the input buffer of P10 and P11 in 1-bit units.

PIM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 12-19. Format of Port Input Mode Registers 1 (PIM1)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM1	0	0Note	0Note	0Note	0Note	0	PIM11	PIM10	F0041H	00H	R/W

PIM1n	P1n pin input buffer selection (n = 0, 1)
0	Normal input buffer
1	TTL input buffer

Note Be sure to clear bits 3 to 6 to “0”.

(17) Port output mode registers 1, 8 (POM1, POM8)

These registers set the output mode of P10 to P12, and P82 in 1-bit units.

POM1 and POM8 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 12-20. Format of Port Output Mode Registers 1, 8 (POM1, POM8)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM1	0	0Note	0Note	0Note	0Note	POM12	POM11	POM10	F0051H	00H	R/W
POM8	0	0	0	0	0	POM82	0	0	F0058H	00H	R/W

POMmn	Pmn pin output mode selection (m = 1, 8; n = 0 to 2)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

Note Be sure to clear bits 3 to 6 to “0”.

(18) Port mode registers 1, 5, 8 (PM1, PM5, PM8)

These registers set input/output of ports 1, 5, and 8 in 1-bit units.

When using the P10/SCK20/SCL20, P11/SI20/SDA20/RxD2/INTP6, P12/SO20/TxD2/TO02, P51/TxD3/SEG38, and P82/TxD0 pins for serial data output or serial clock output, clear the PM10, PM11, PM12, PM51, and PM82 bits to 0, and set the output latches of P10, P11, P12, P51, and P82 to 1.

When using the P10/SCK20/SCL20, P11/SI20/SDA20/RxD2/INTP6, P50/RxD3/SEG39, and P81/RxD0/INTP9 pins for serial data input or serial clock input, set the PM10, PM11, PM50, and PM81 bits to 1. At this time, the output latches of P10, P11, P50, and P81 may be set to 0 or 1.

PM1, PM5, and PM8 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 12-21. Format of Port Mode Registers 1, 5, and 8 (PM1, PM5, PM8)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM1	1	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM8	1	1	1	1	1	PM82	PM81	PM80 Note	FFF28H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 1, 5, 8; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note Be sure to clear bits PM80 of PM8 to “0” after reset release.

12.4 Operation stop mode

Each serial interface of serial array unit has the operation stop mode.

In this mode, serial communication cannot be executed, thus reducing the power consumption.

In addition, the P10/SCK20/SCL20, P11/SI20/SDA20/RxD2/INTP6, P12/SO20/TxD2/TO02, P50/RxD3/SEG39, P51/TxD3/SEG38, P81/RxD0/INTP9, and P82/TxD0 pins can be used as ordinary port pins in this mode.

12.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

PER0 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.

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 12-22. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAUm to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	RTCEN		ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
	×	0	×	×	0/1	0/1	×	×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Caution 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 (except for input switch control register (ISC), noise filter enable register (NFEN0), port input mode register (PIM1), port output mode registers (POM1, POM8), port mode registers (PM1, PM5, PM8), and port registers (P1, P5, P8)).

Remark m: Unit number (m = 0, 1)
□: Setting disabled (set to the initial value)
×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

12.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 12-23. Each Register Setting When Stopping the Operation by Channels (1/2)

(a) Serial Channel Enable Status Register m (SEm) …

This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

SE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	SE02 Note	SE01 0/1	SE00 0/1

0: Operation stops

- The SE0 register is a read-only status register, whose operation is stopped by using the ST0 register.

With a channel whose operation is stopped, the value of CKO0n of the SO0 register can be set by software.

SE1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SE13 0/1	SE12 0/1	SE11 0/1	SE10 0/1

0: Operation stops

- The SE1 register is a read-only status register, whose operation is stopped by using the ST1 register.

With a channel whose operation is stopped, the value of CKO1n of the SO1 register can be set by software.

(b) Serial channel stop register m (STm) …

This register is a trigger register that is used to enable stopping communication/count by each channel.

ST0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	ST02 Note	ST01 0/1	ST00 0/1

1: Clears SE0n to 0 and stops the communication operation

* Because ST0n is a trigger bit, ST0n is cleared immediately when SE0n = 0.

ST1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	ST13 0/1	ST12 0/1	ST11 0/1	ST10 0/1

1: Clears SE1n to 0 and stops the communication operation

* Because ST1n is a trigger bit, ST1n is cleared immediately when SE1n = 0.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

□ : Setting disabled (set to the initial value), 0/1: Set to 0 or 1 depending on the usage of the user

Note This is a bit used for extended SFR (3rd SFR).

See **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE.**

Figure 12-23. Each Register Setting When Stopping the Operation by Channels (2/2)

(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.

SOE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 Note	0	SOE00 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SO0n value of the SO0 register can be set by software.

SOE1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12 0/1	0	SOE10 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SO1n value of the SO1 register can be set by software.

(d) Serial output register m (SOm) …

This register is a buffer register for serial output of each channel.

SO0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	CKO02 Note	1	1	0	0	0	0	1	SO02 Note	1	SO00 0/1

1: Serial data output value is “1”

* When using pins corresponding to each channel as port function pins, set the corresponding CKO0n and SO0n bits to “1”.

SO1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	CKO10 0/1	0	0	0	0	1	SO12 0/1	1	SO10 0/1

1: Serial clock output value is “1” 1: Serial data output value is “1”

* When using pins corresponding to each channel as port function pins, set the corresponding CKO10 and SO1n bits to “1”.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
: Setting disabled (set to the initial value), 0/1: Set to 0 or 1 depending on the usage of the user

Note This is a bit used for extended SFR (3rd SFR).
See **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE.**

12.5 Operation of 3-Wire Serial I/O (CSI20) Communication

This is a clocked communication function that uses three lines: serial clock ($\overline{SCK}$) 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
- Level setting of transmit/receive data

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Caution CSI10 (channel 2 of unit 0) is dedicated for the extended SFR (3rd SFR).
See CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE for details.

The channel supporting 3-wire serial I/O (CSI20) is channel 0 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	–	UART0	–
	1	–		–
	2	CSI10 (dedicated to the extended SFR (3rd SFR) interface)	–	–
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

3-wire serial I/O (CSI20) performs the following six types of communication operations.

- Master transmission (See **12.5.1**.)
- Master reception (See **12.5.2**.)
- Master transmission/reception (See **12.5.3**.)
- Slave transmission (See **12.5.4**.)
- Slave reception (See **12.5.5**.)
- Slave transmission/reception (See **12.5.6**.)

12.5.1 Master transmission

Master transmission is that the 78K0R/Lx3-M microcontrollers output a transfer clock and transmit data to another device.

3-Wire Serial I/O	CSI20
Target channel	Channel 0 of SAU1
Pins used	$\overline{\text{SCK20}}$, SO20
Interrupt	INTCSI20
	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	Max. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz]Note f_{CLK}: System clock frequency
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

(1) Register setting

Figure 12-24. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI20)

(a) Serial output register 1 (SO1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10 0/1	0	0	0	0	1	SO12 ×	1	SO10 0/1

Communication starts when this bit is 1 if the data phase is forward (CKP10 = 0). If the phase is reversed (CKP10 = 1), communication starts when these bits are 0.

(b) Serial output enable register 1 (SOE1) … 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
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12 ×	0	SOE10 0/1

(c) Serial channel start register 1 (SS1) … 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
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13 ×	SS12 ××	SS11 ×	SS10 0/1

(d) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10 0/1	CCS10 0	0	0	0	0	0	STS10 0	0	SIS10 0	1	0	0	MD102 0	MD101 0	MD100 0/1

Operation mode of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10 1	RXE10 0	DAP10 0/1	CKP10 0/1	0	EOC10 0	PTC101 0	PTC100 0	DIR10 0/1	0	SLC101 0	SLC100 0	0	DLS102 1	DLS101 1	DLS100 0/1

(f) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

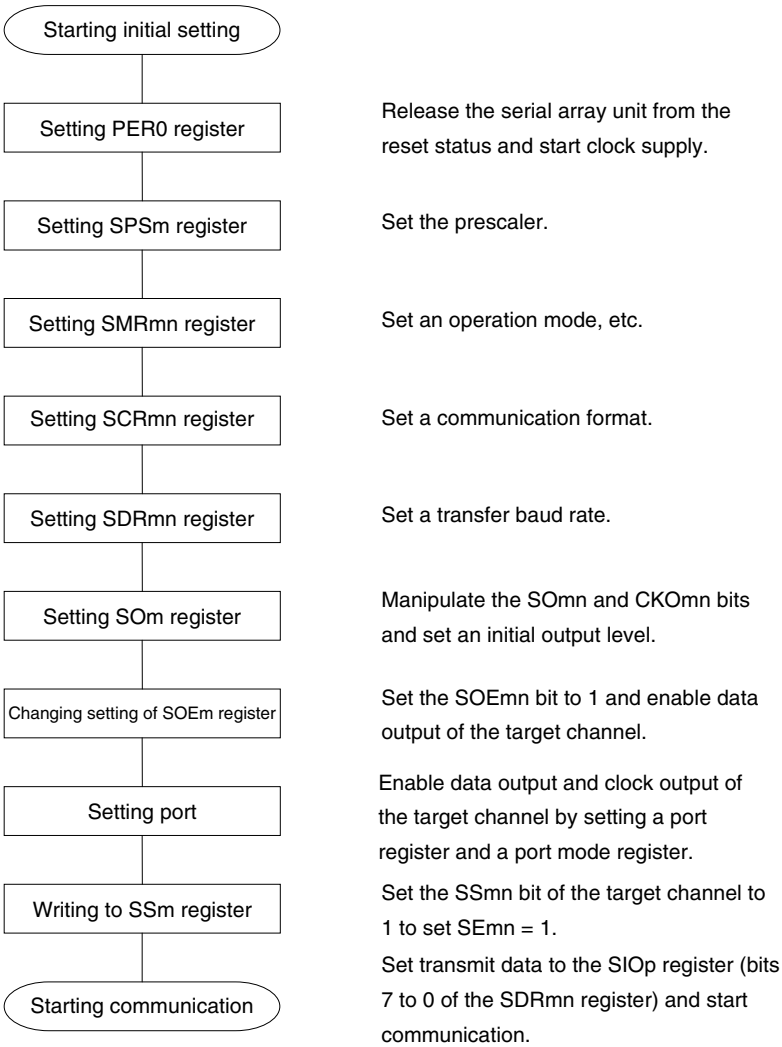
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting							0	Transmit data setting							

SIO20

Remark □: 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) Operation procedure

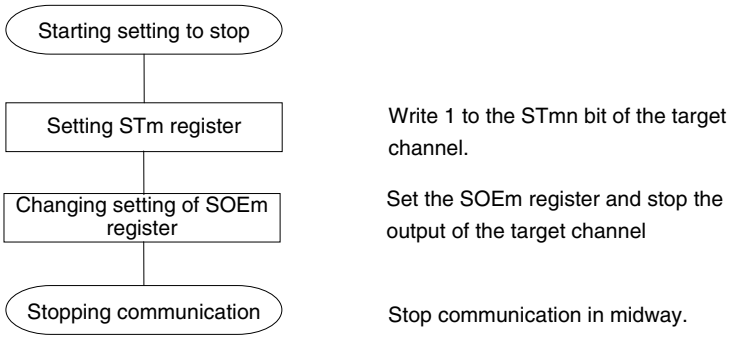
Figure 12-25. Initial Setting Procedure for Master Transmission



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

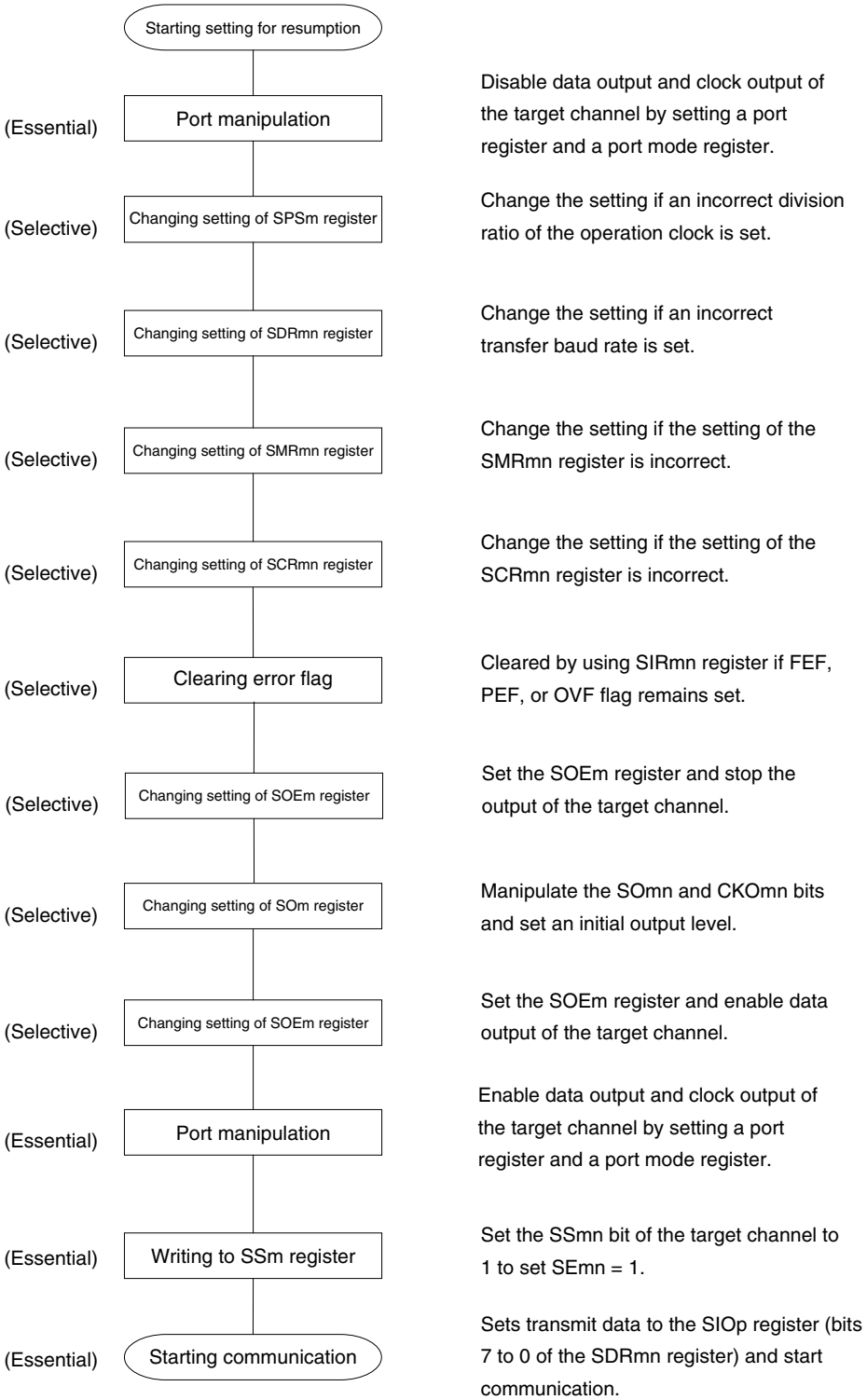
Figure 12-26. Procedure for Stopping Master Transmission



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 12-27 Procedure for Resuming Master Transmission**).

2. m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

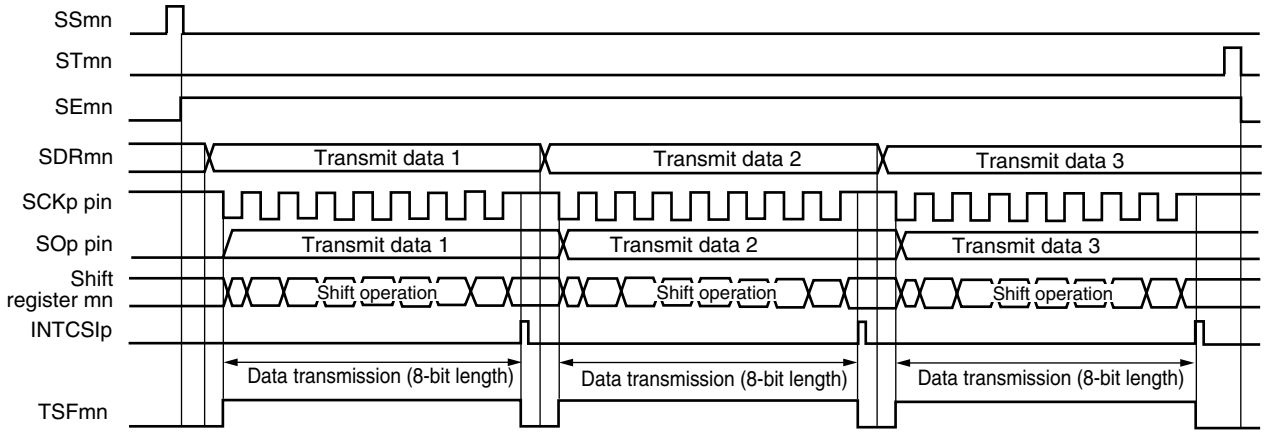
Figure 12-27. Procedure for Resuming Master Transmission



Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

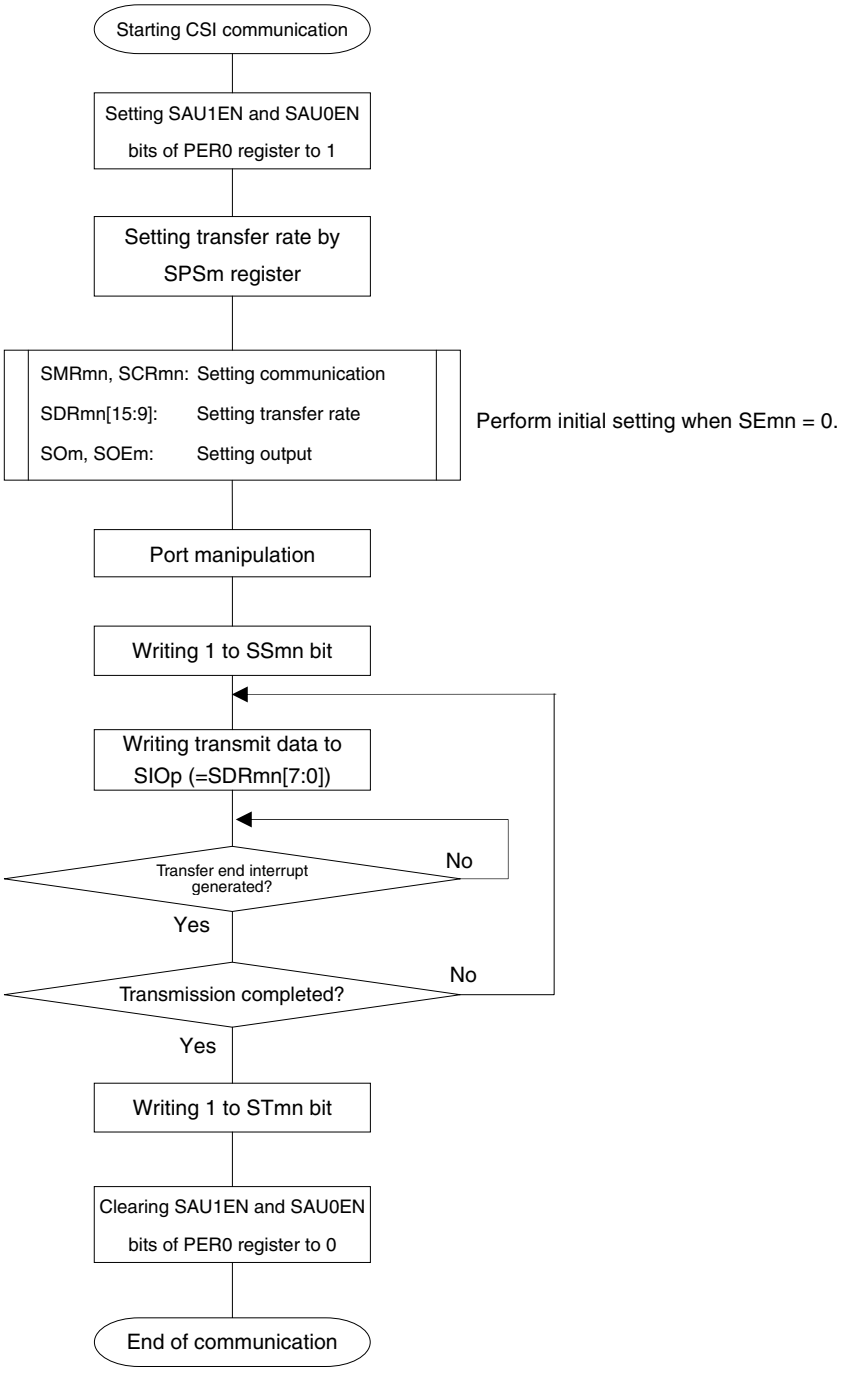
(3) Processing flow (in single-transmission mode)

Figure 12-28. Timing Chart of Master Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

Figure 12-29. Flowchart of Master Transmission (in Single-Transmission Mode)

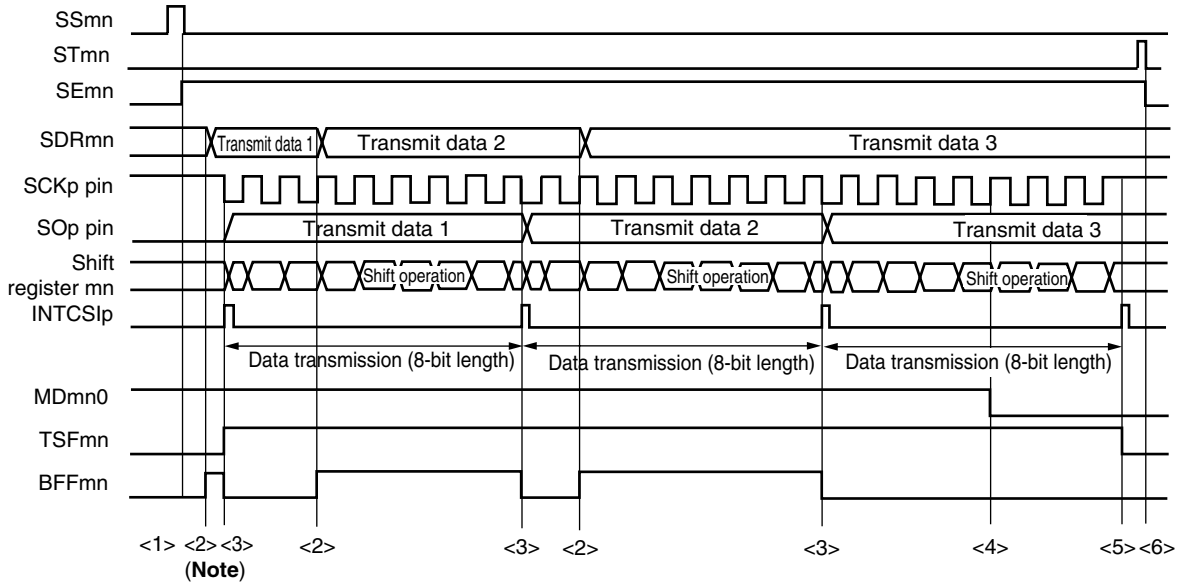


Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

(4) Processing flow (in continuous transmission mode)

Figure 12-30. Timing Chart of Master Transmission (in Continuous Transmission Mode)

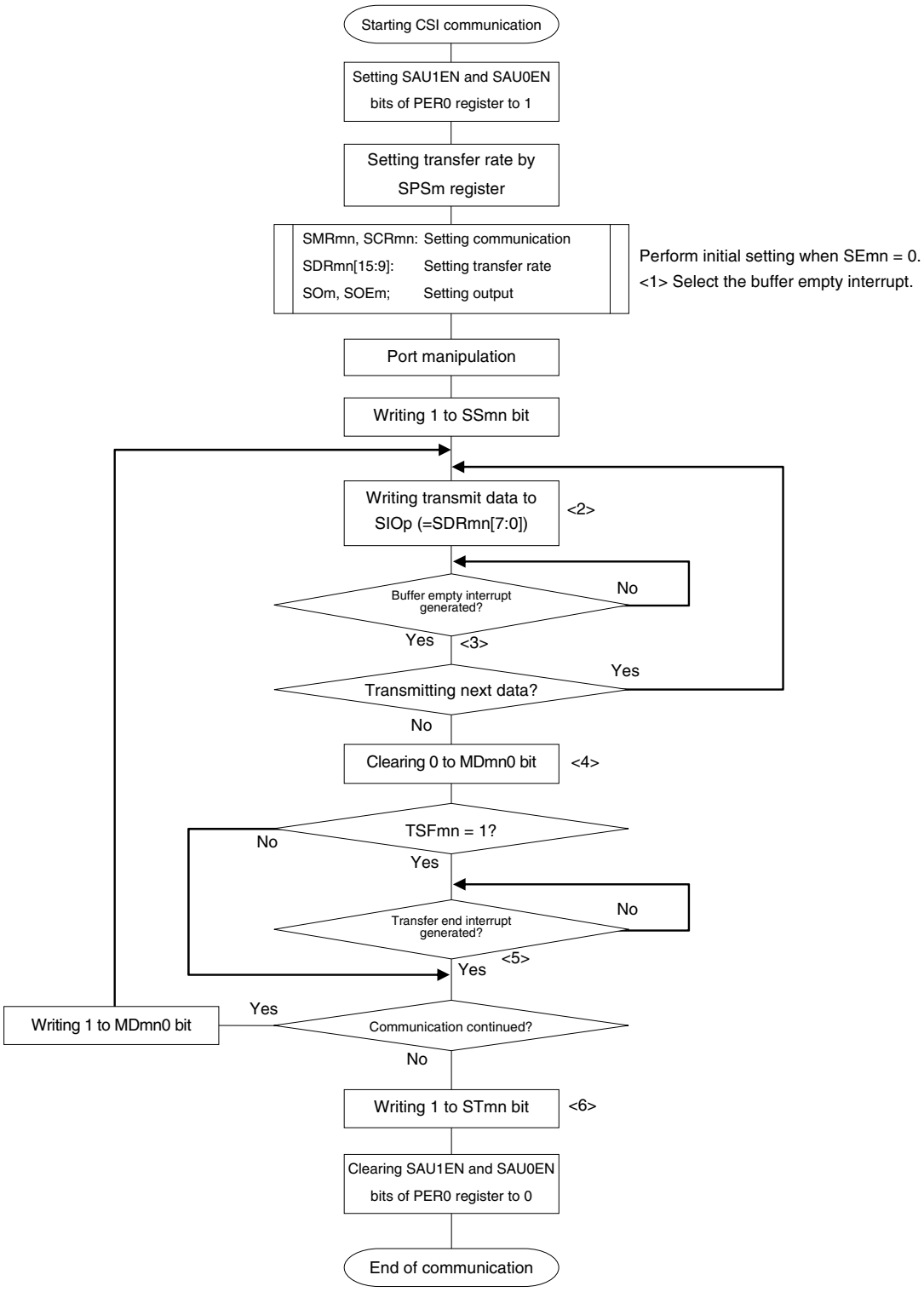


Note When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

Caution The MDmn0 bit 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 = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

Figure 12-31. Flowchart of Master Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-30 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

2. m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

12.5.2 Master reception

Master reception is that the 78K0R/Lx3-M microcontrollers output a transfer clock and receive data from other device.

3-Wire Serial I/O	CSI20
Target channel	Channel 0 of SAU1
Pins used	$\overline{SCK20}$, SI20
Interrupt	INTCSI20
	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. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz]Note f_{CLK}: System clock frequency
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

(1) Register setting

Figure 12-32. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI20)

(a) Serial output register 1 (SO1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10 0/1	0	0	0	0	1	SO12 ×	1	SO10 ×

Communication starts when this bit is 1 if the data phase is forward (CKP10 = 0). If the phase is reversed (CKP10 = 1), communication starts when these bits are 0.

(b) Serial output enable register 1 (SOE1) … Unused in this mode.

	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	SOE12 ×	0	SOE10 ×

(c) Serial channel start register 1 (SS1) … 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
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13 ×	SS12 ×	SS11 ×	SS10 0/1

(d) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10 0/1	CCS10 0	0	0	0	0	0	STS10 0	0	SIS100 0	1	0	0	MD102 0	MD101 0	MD100 0

Operation mode of channel 0
0: Transfer end interrupt

(e) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10 0	RXE10 1	DAP10 0/1	CKP10 0/1	0	EOC10 0	PTC101 0	PTC100 0	DIR10 0/1	0	SLC101 0	SLC100 0	0	DLS102 1	DLS101 1	DLS100 0/1

(f) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

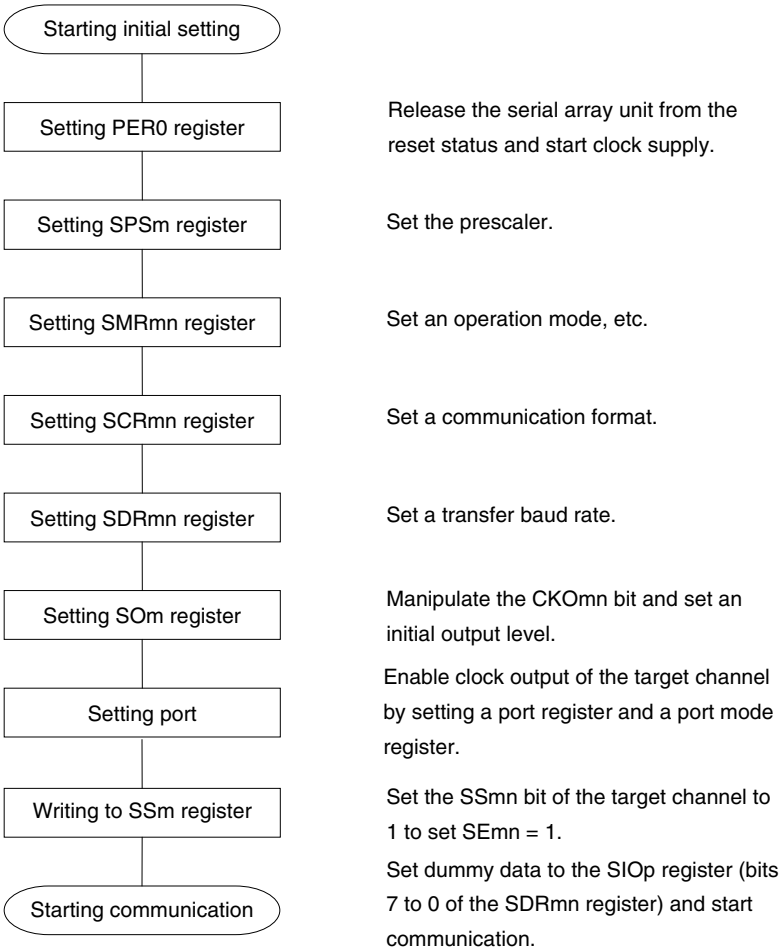
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting							0	Receive data register (Write FFH as dummy data.)							

SIO20

Remark □: 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

(2) Operation procedure

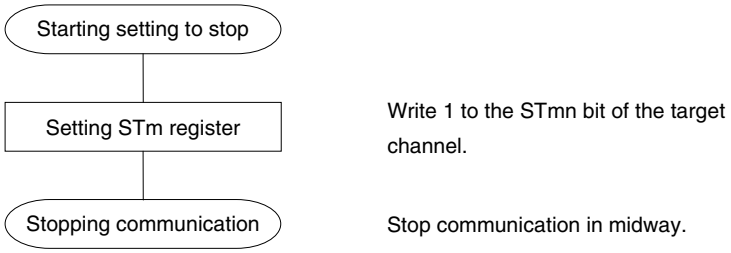
Figure 12-33. Initial Setting Procedure for Master Reception



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

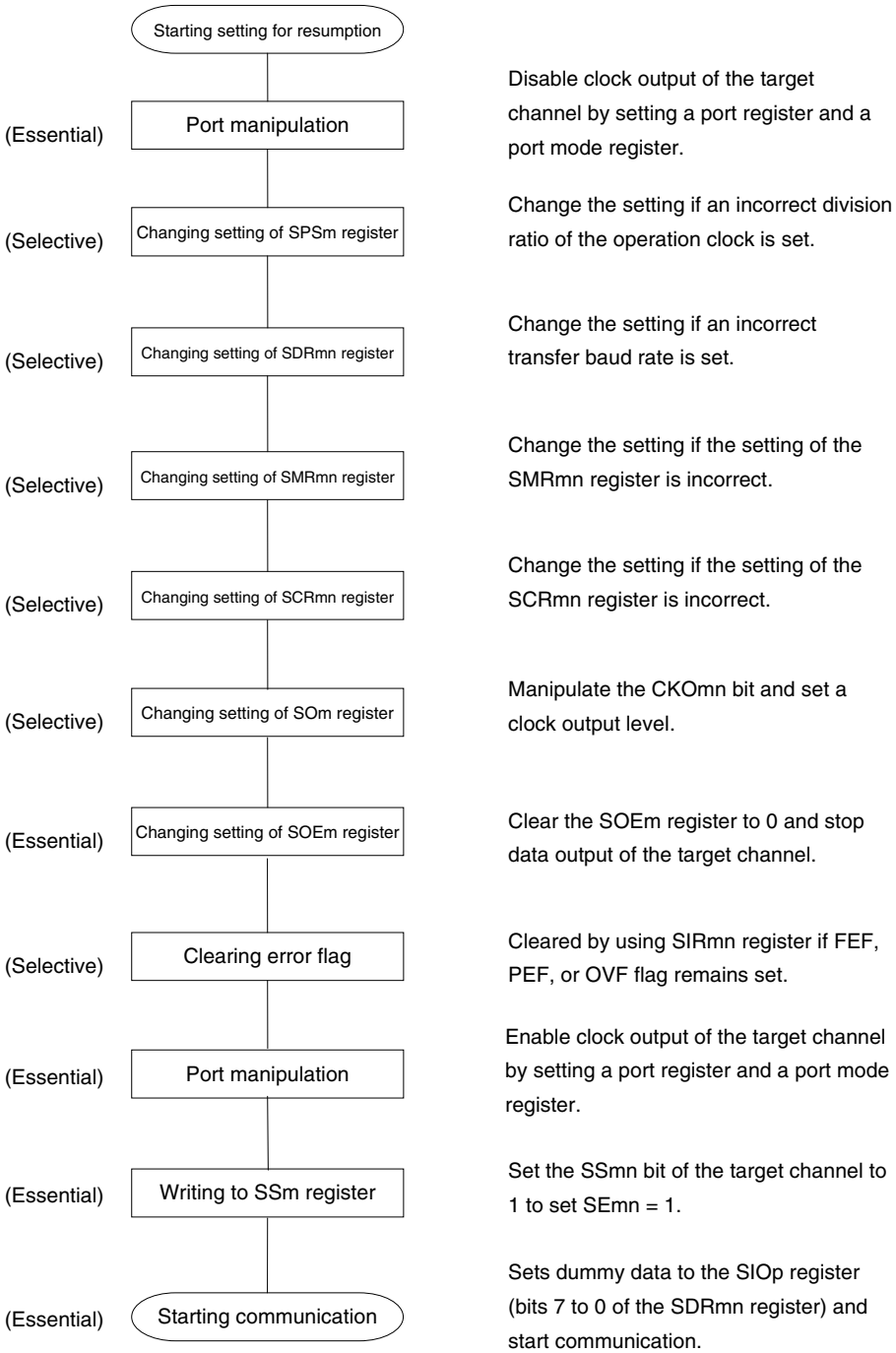
Figure 12-34. Procedure for Stopping Master Reception



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 12-35 Procedure for Resuming Master Reception**).

2. m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

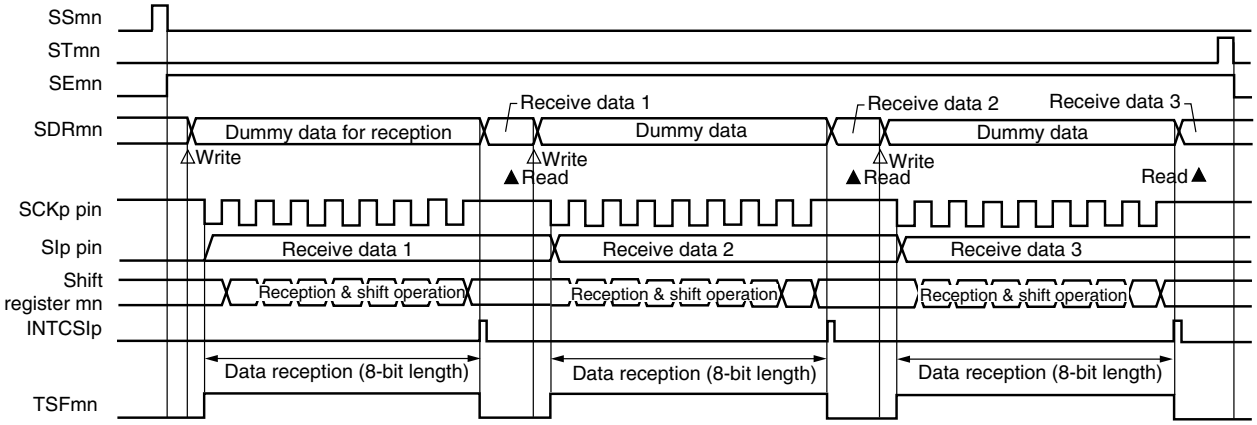
Figure 12-35. Procedure for Resuming Master Reception



Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

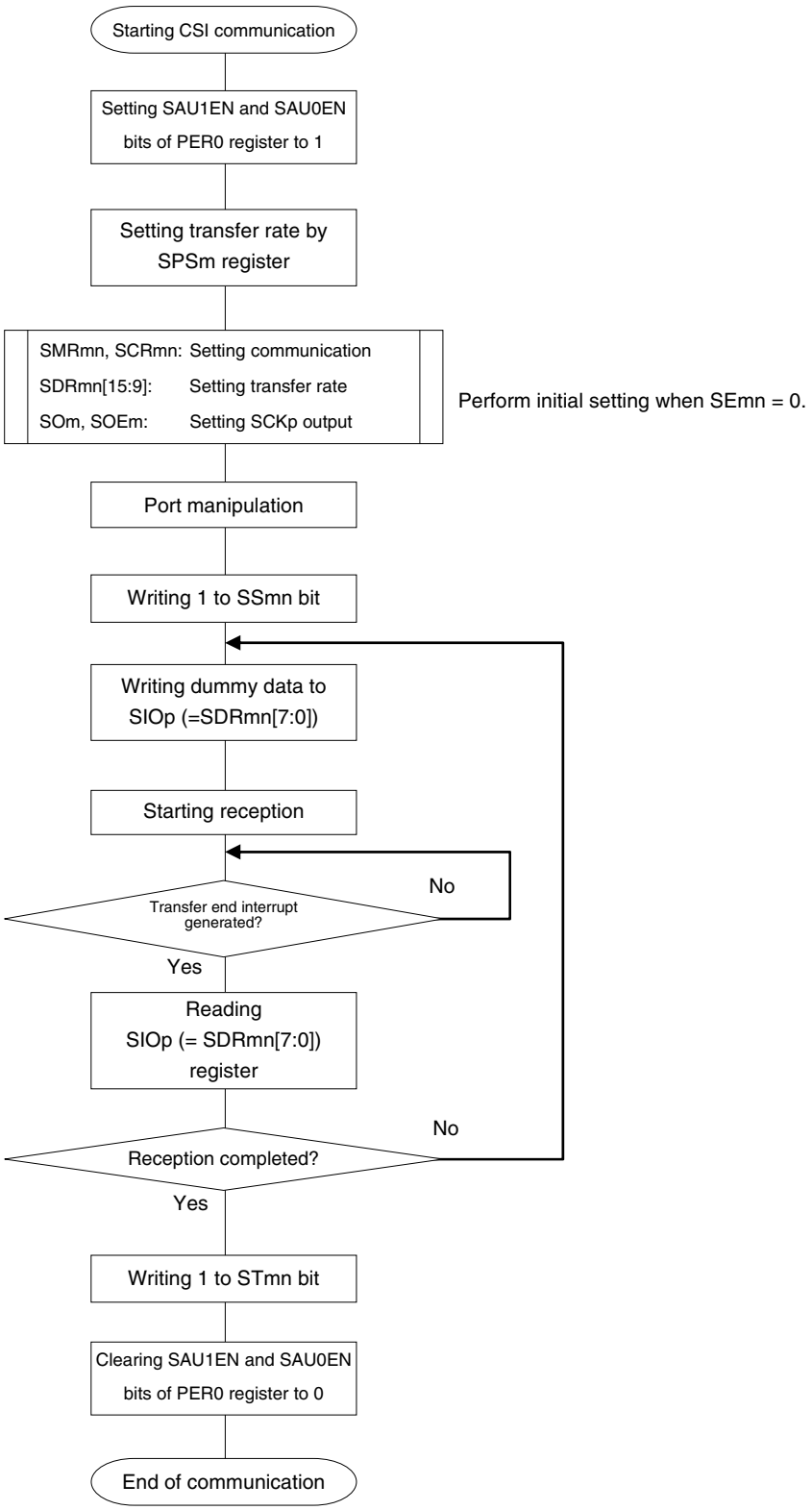
(3) Processing flow (in single-reception mode)

Figure 12-36. Timing Chart of Master Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

Figure 12-37. Flowchart of Master Reception (in Single-Reception Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

12.5.3 Master transmission/reception

Master transmission/reception is that the 78K0R/Lx3-M microcontrollers output a transfer clock and transmit/receive data to/from other device.

3-Wire Serial I/O	CSI20
Target channel	Channel 0 of SAU1
Pins used	$\overline{\text{SCK20}}$, SI20, SO20
Interrupt	INTCSI20
	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. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz]Note f_{CLK}: System clock frequency
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

(1) Register setting

Figure 12-38. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI20)

(a) Serial output register 1 (SO1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10 0/1	0	0	0	0	1	SO12 ×	1	SO10 0/1

Communication starts when these bits are 1 if the data phase is forward (CKP10 = 0). If the phase is reversed (CKP10 = 1), communication starts when these bits are 0.

(b) Serial output enable register 1 (SOE1) … 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
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12 ×	0	SOE10 0/1

(c) Serial channel start register 1 (SS1) … 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
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13 ×	SS12 ×	SS11 ×	SS10 0/1

(d) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10 0/1	CCS10 0	0	0	0	0	0	STS10 0	0	SIS100 0	1	0	0	MD102 0	MD101 0	MD100 0/1

Operation mode of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10 1	RXE10 1	DAP10 0/1	CKP10 0/1	0	EOC10 0	PTC101 0	PTC100 0	DIR10 0/1	0	SLC101 0	SLC100 0	0	DLS102 1	DLS101 1	DLS100 0/1

(f) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

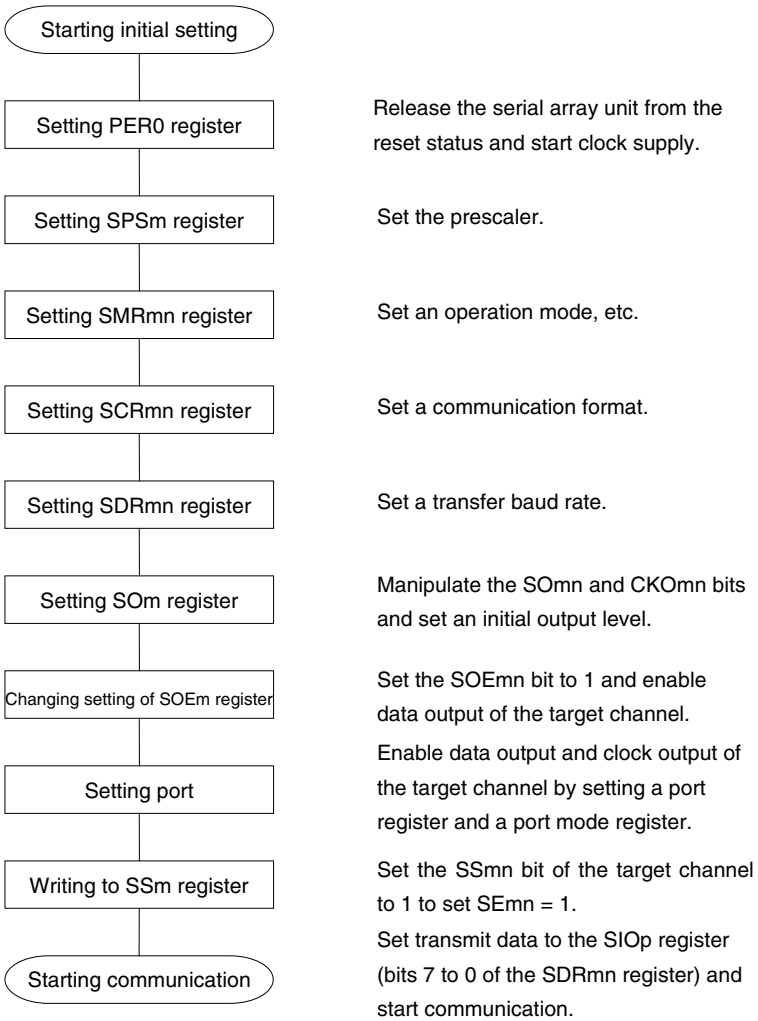
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting							0	Transmit data setting/receive data register							

SIO20

Remark □: 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

(2) Operation procedure

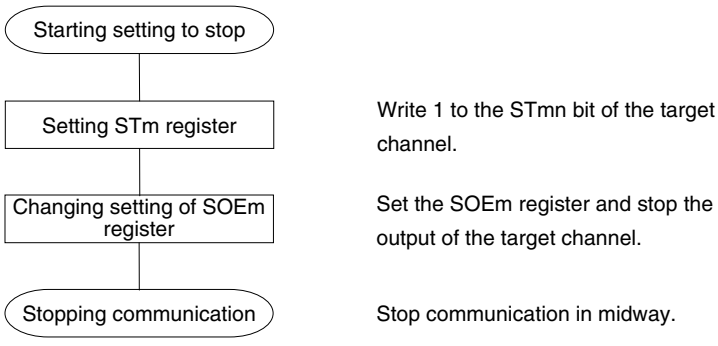
Figure 12-39. Initial Setting Procedure for Master Transmission/Reception



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

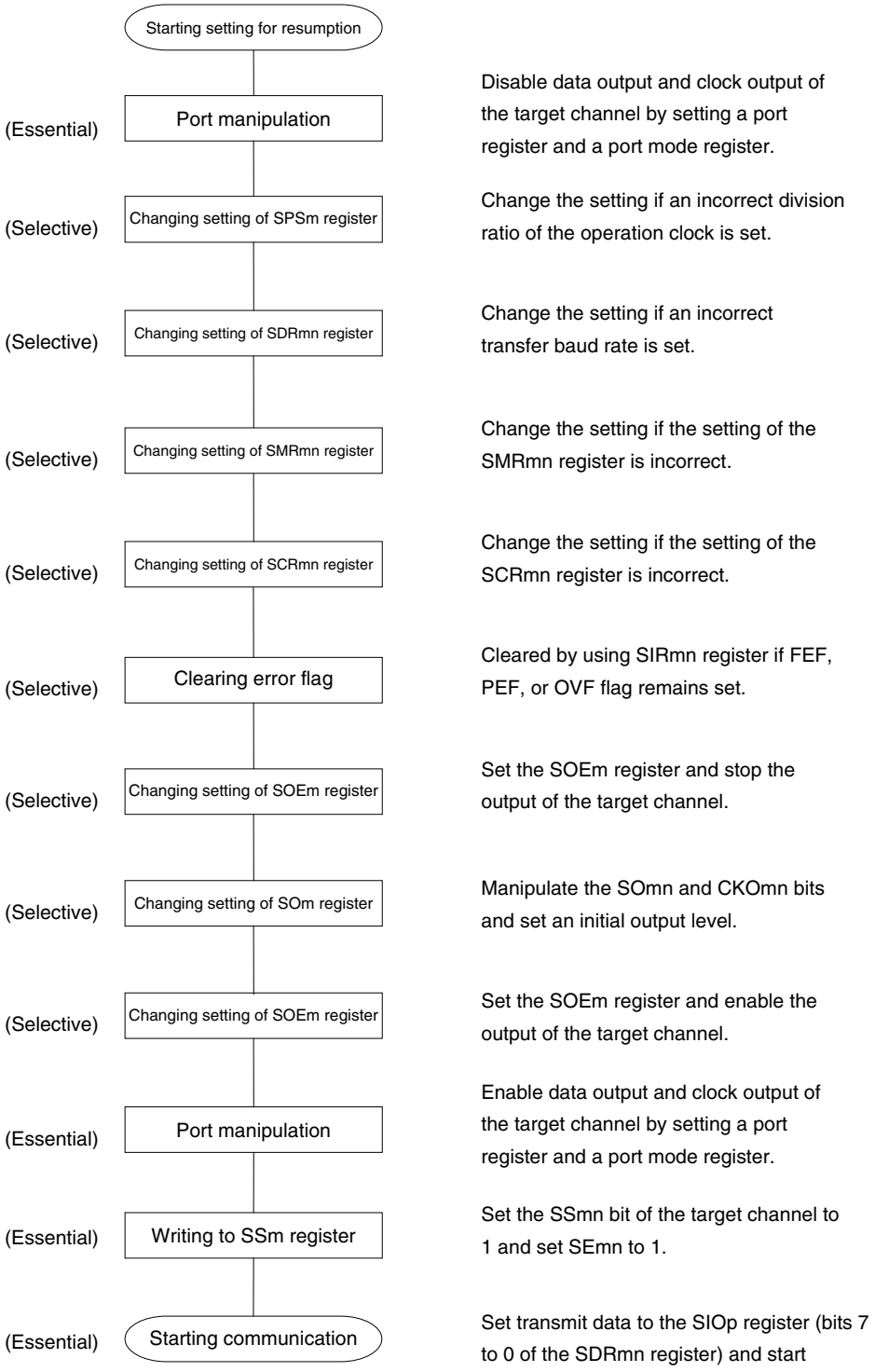
Figure 12-40. Procedure for Stopping Master Transmission/Reception



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 12-41 Procedure for Resuming Master Transmission/Reception**).

2. m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

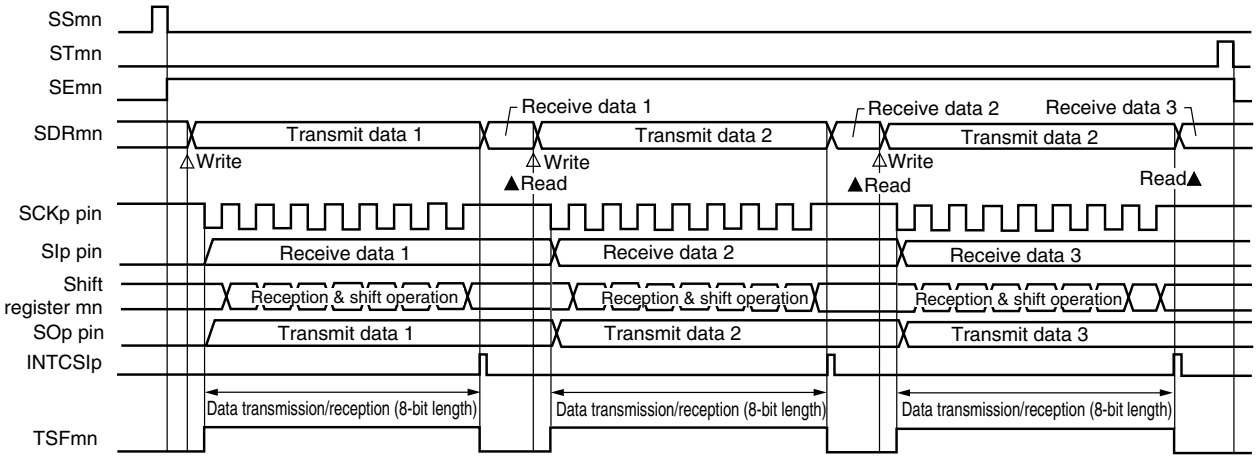
Figure 12-41. Procedure for Resuming Master Transmission/Reception



Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

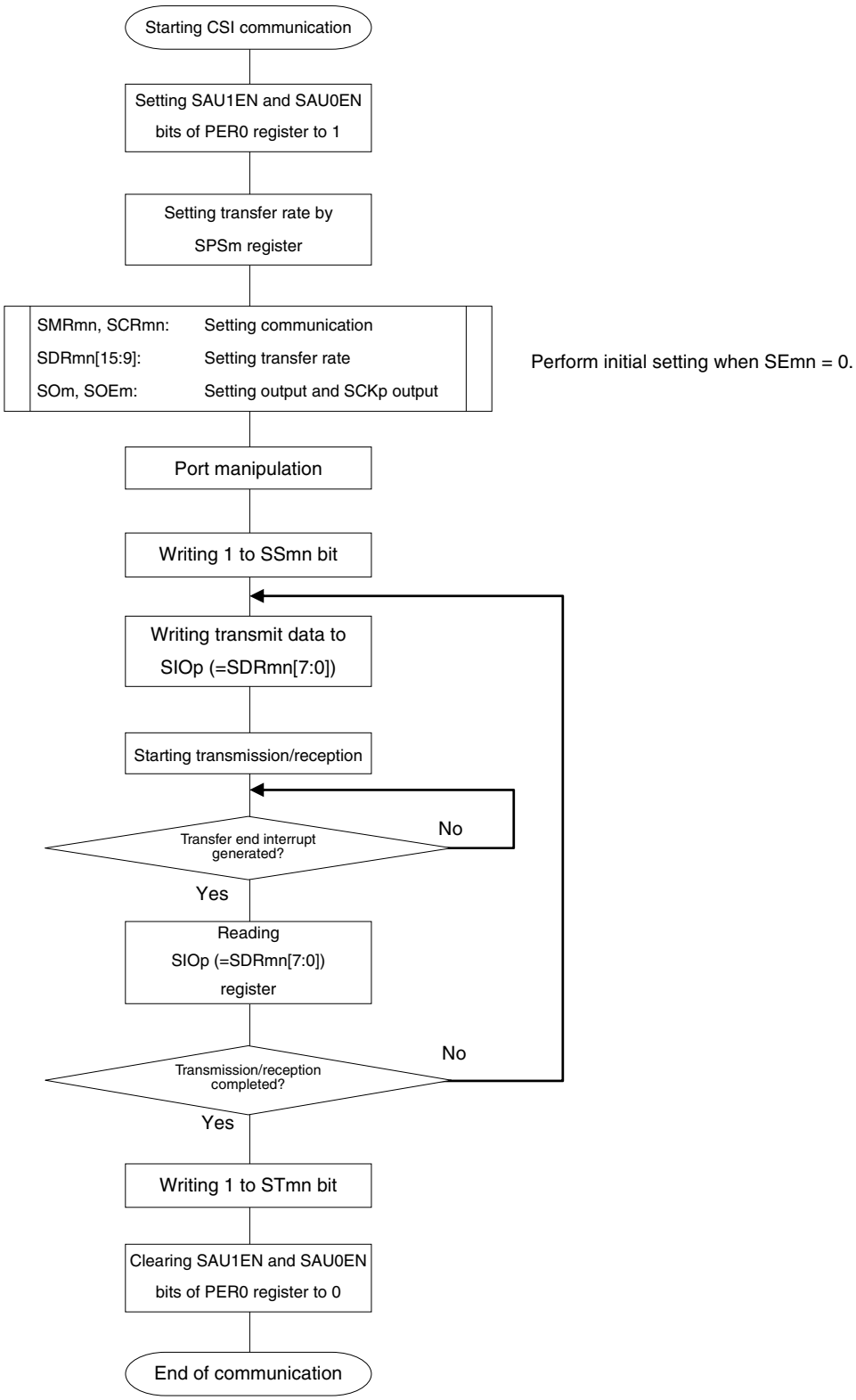
(3) Processing flow (in single-transmission/reception mode)

Figure 12-42. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

Figure 12-43. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)

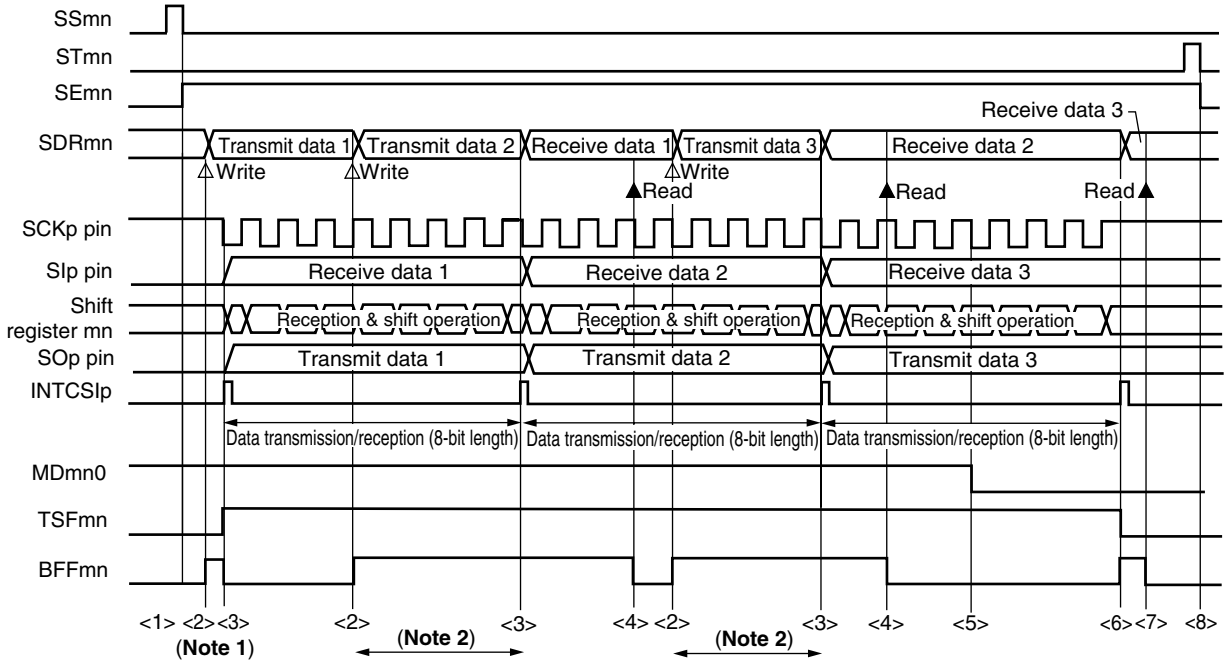


Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

(4) Processing flow (in continuous transmission/reception mode)

Figure 12-44. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 1. When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

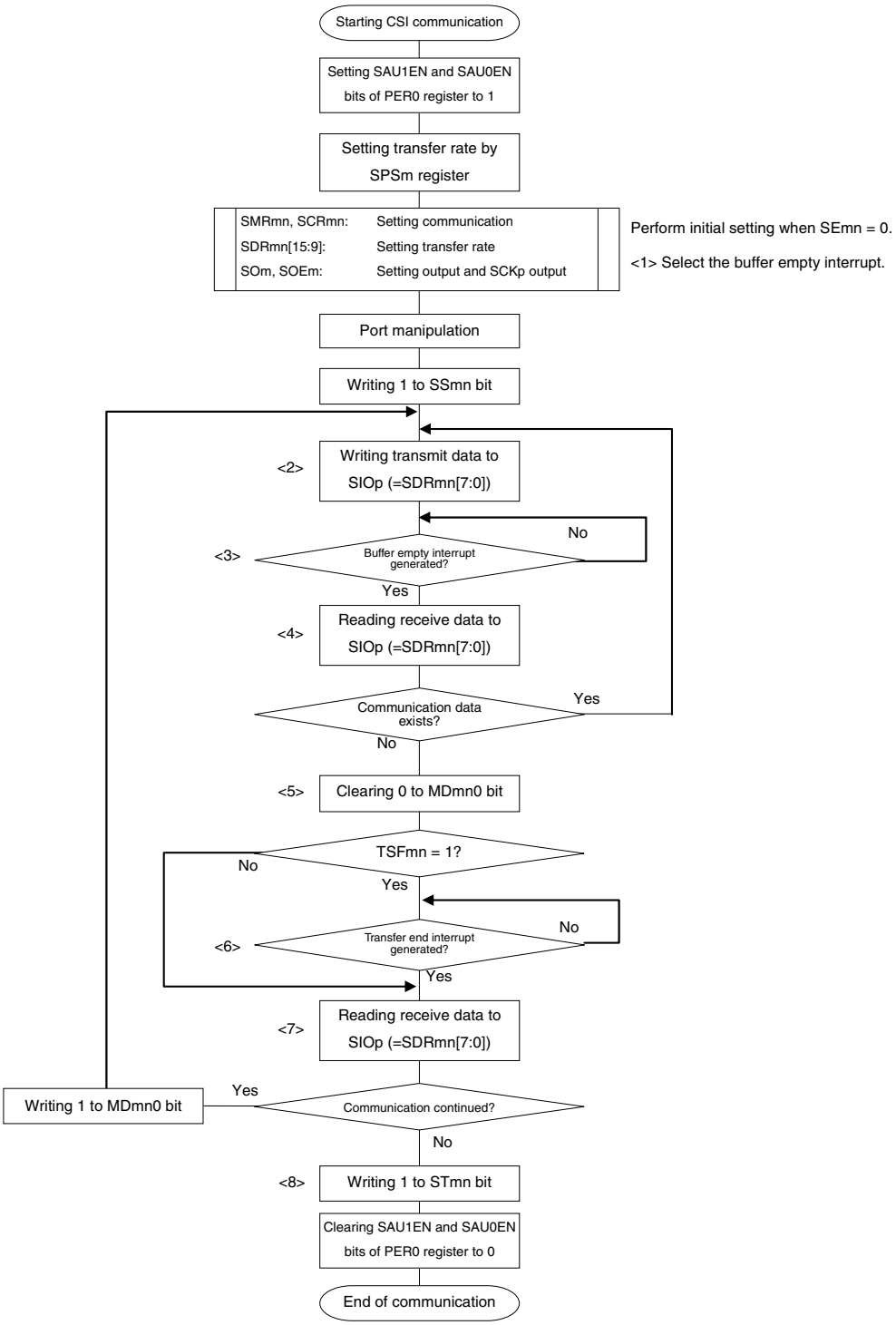
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-45 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

Figure 12-45. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-44 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

12.5.4 Slave transmission

Slave transmission is that the 78K0R/Lx3-M microcontrollers transmit data to another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI20
Target channel	Channel 0 of SAU1
Pins used	$\overline{SCK20}$, SO20
Interrupt	INTCSI20
	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 [MHz]Notes 1, 2
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Notes 1. Because the external serial clock input to pin $\overline{SCK20}$ is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [MHz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel

2. m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

(1) Register setting

Figure 12-46. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI20)

(a) Serial output register 1 (SO1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10 ×	0	0	0	0	1	SO12 ×	1	SO10 0/1

(b) Serial output enable register 1 (SOE1) … 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
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12 ×	0	SOE10 0/1

(c) Serial channel start register 1 (SS1) … 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
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13 ×	SS12 ×	SS11 ×	SS10 0/1

(d) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10 0/1	CCS10 1	0	0	0	0	0	STS10 0	0	SIS100 0	1	0	0	MD102 0	MD101 0	MD100 0/1

Operation mode of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10 1	RXE10 0	DAP10 0/1	CKP10 0/1	0	EOC10 0	PTC101 0	PTC100 0	DIR10 0/1	0	SLC101 0	SLC100 0	0	DLS102 1	DLS101 1	DLS100 0/1

(f) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

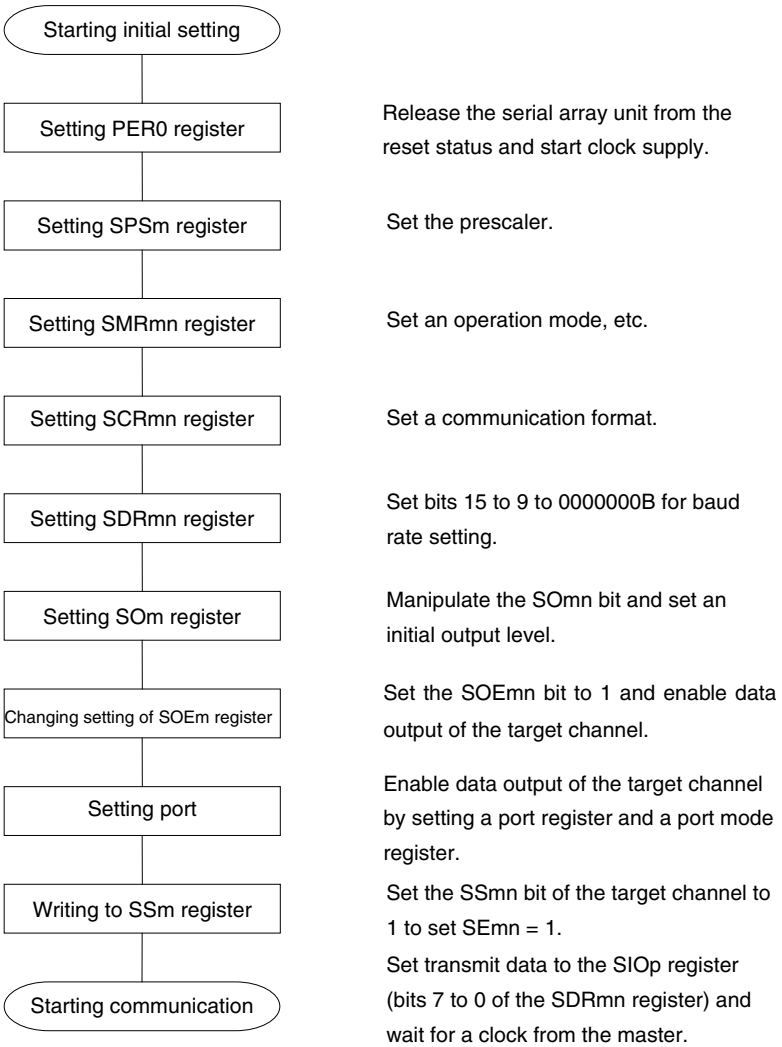
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting							0	Transmit data setting							

SIO20

Remark □: 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

(2) Operation procedure

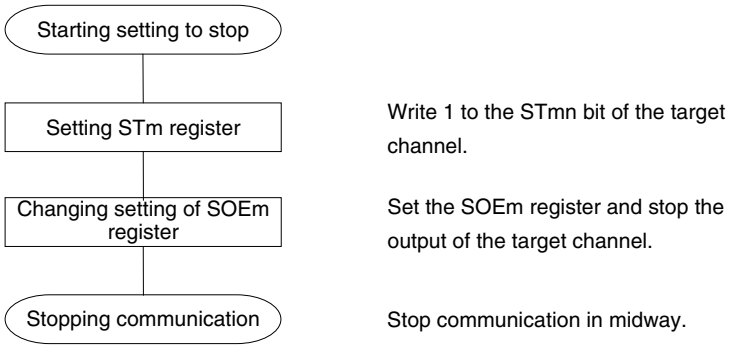
Figure 12-47. Initial Setting Procedure for Slave Transmission



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

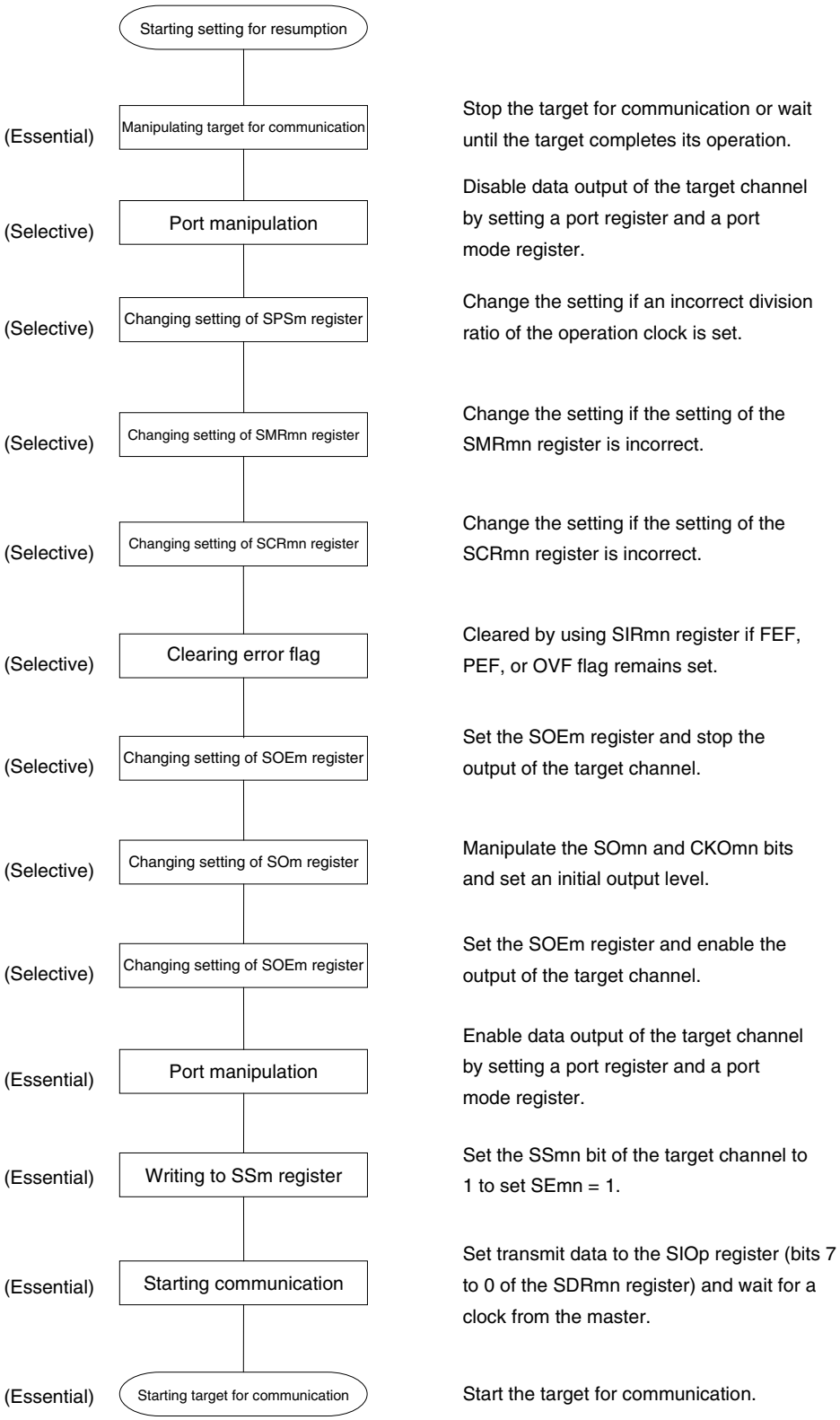
Figure 12-48. Procedure for Stopping Slave Transmission



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 12-49 Procedure for Resuming Slave Transmission**).

2. m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

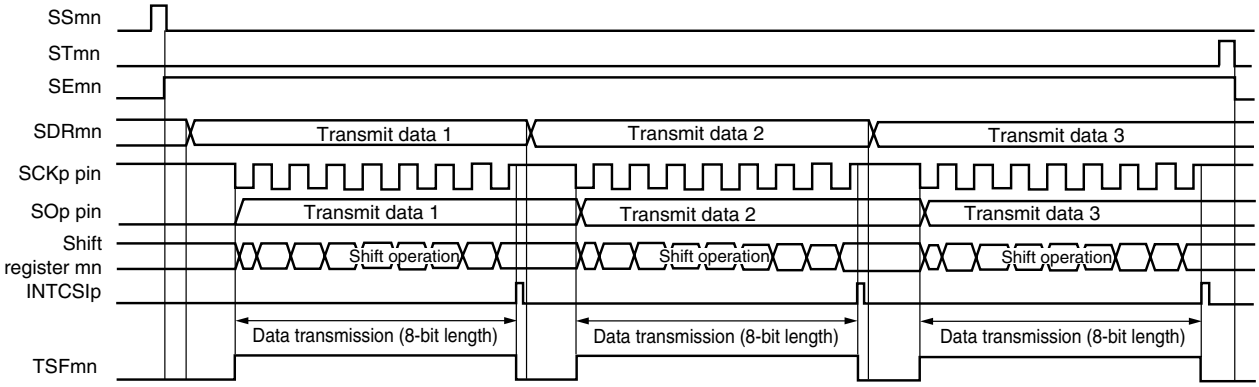
Figure 12-49. Procedure for Resuming Slave Transmission



Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

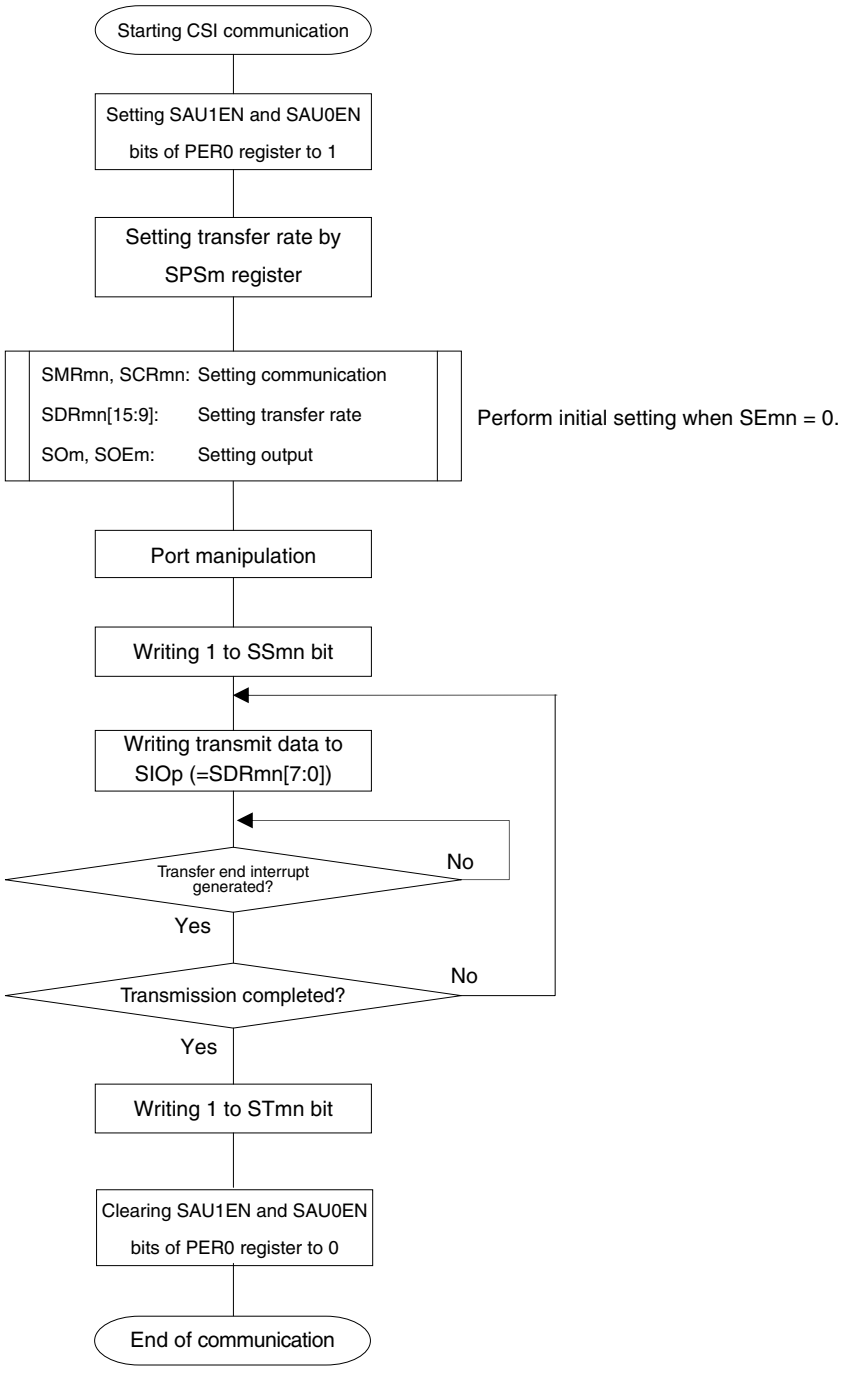
(3) Processing flow (in single-transmission mode)

Figure 12-50. Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

Figure 12-51. Flowchart of Slave Transmission (in Single-Transmission Mode)

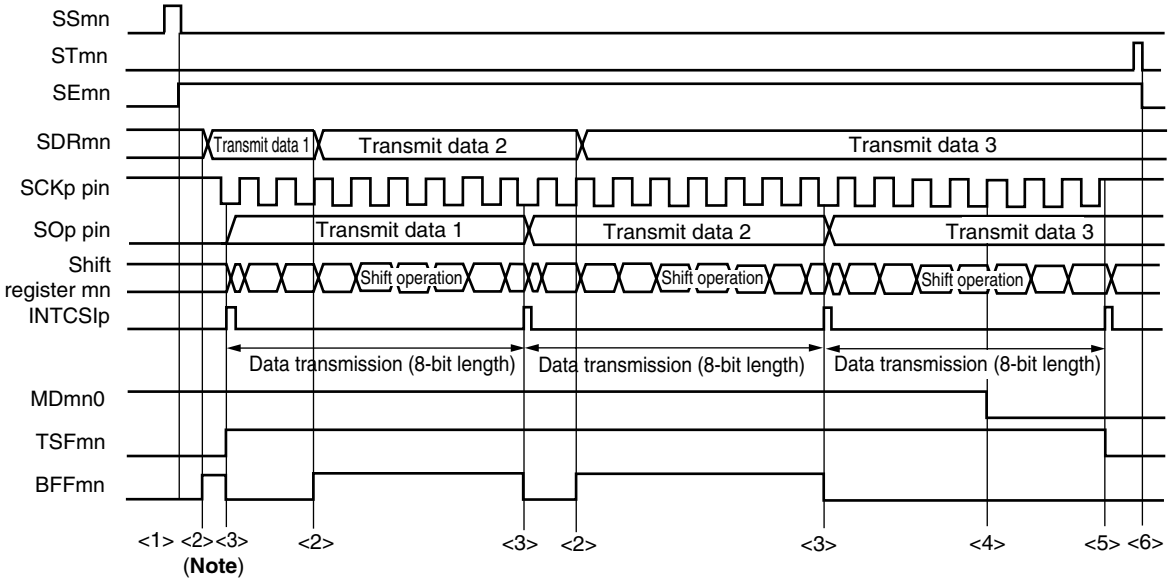


Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

(4) Processing flow (in continuous transmission mode)

Figure 12-52. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

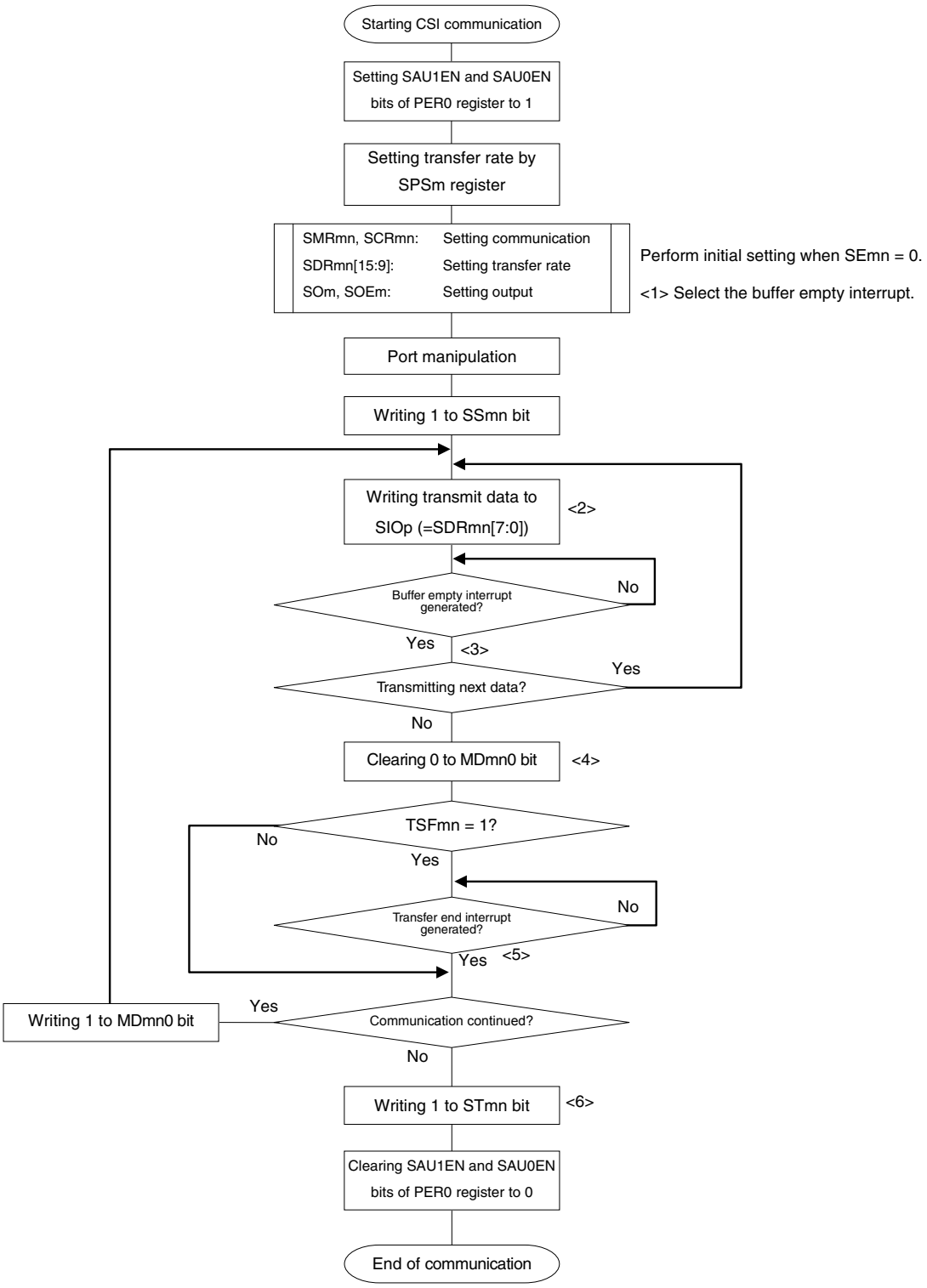


Note When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

Caution The MDmn0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

Figure 12-53. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-52 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

2. m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

12.5.5 Slave reception

Slave reception is that the 78K0R/Lx3-M microcontrollers receive data from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI20
Target channel	Channel 0 of SAU1
Pins used	$\overline{SCK20}$, SI20
Interrupt	INTCSI20
	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 [MHz]Notes 1, 2
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Notes 1. Because the external serial clock input to pin $\overline{SCK20}$ is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [MHz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel

2. m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

(1) Register setting

Figure 12-54. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI20)

(a) Serial output register 1 (SO1)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10 ×	0	0	0	0	1	SO12 ×	1	SO10 ×

(b) Serial output enable register 1 (SOE1) …Unused in this mode.

	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	SOE12 ×	0	SOE10 ×

(c) Serial channel start register 1 (SS1) … 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
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13 ×	SS12 ×	SS11 ×	SS10 0/1

(d) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10 0/1	CCS10 1	0	0	0	0	0	STS10 0	0	SIS100 0	1	0	0	MD102 0	MD101 0	MD100 0

Operation mode of channel 0
0: Transfer end interrupt

(e) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10 0	RXE10 1	DAP10 0/1	CKP10 0/1	0	EOC10 0	PTC101 0	PTC100 0	DIR10 0/1	0	SLC101 0	SLC100 0	0	DLS102 1	DLS101 1	DLS100 0/1

(f) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

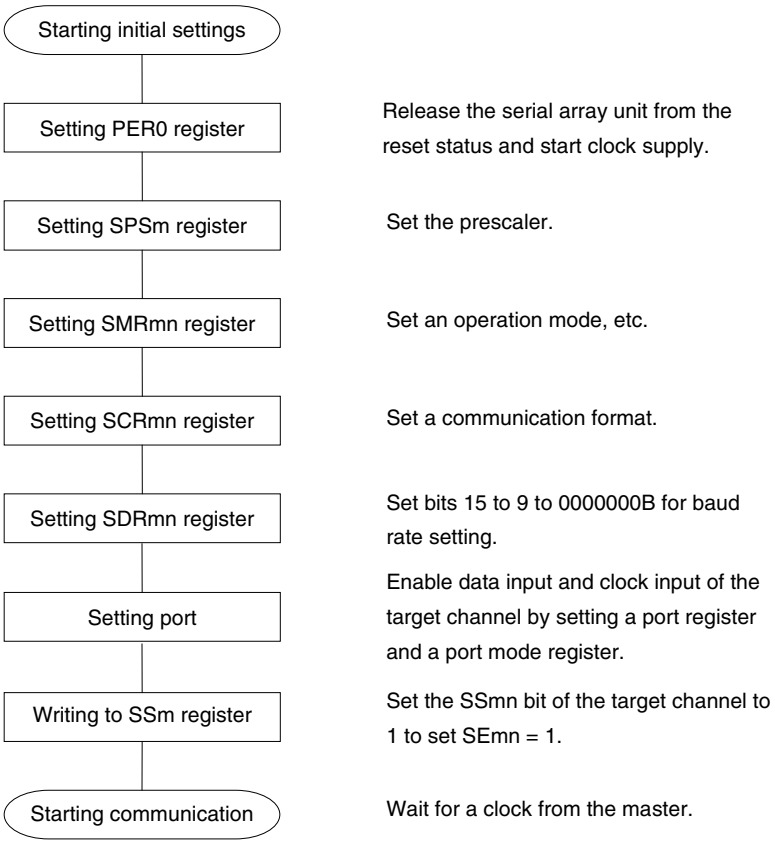
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	0000000 (baud rate setting)							0	Receive data register							

SIO20

Remark □: Setting is fixed in the CSI slave reception mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

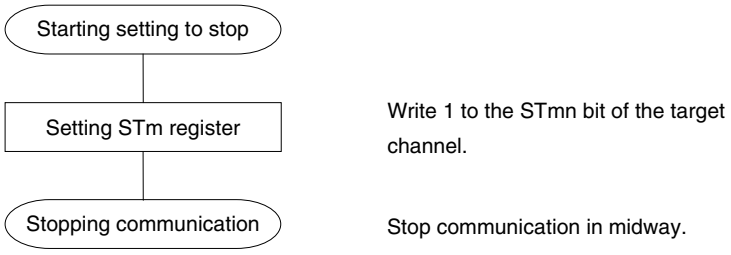
Figure 12-55. Initial Setting Procedure for Slave Reception



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

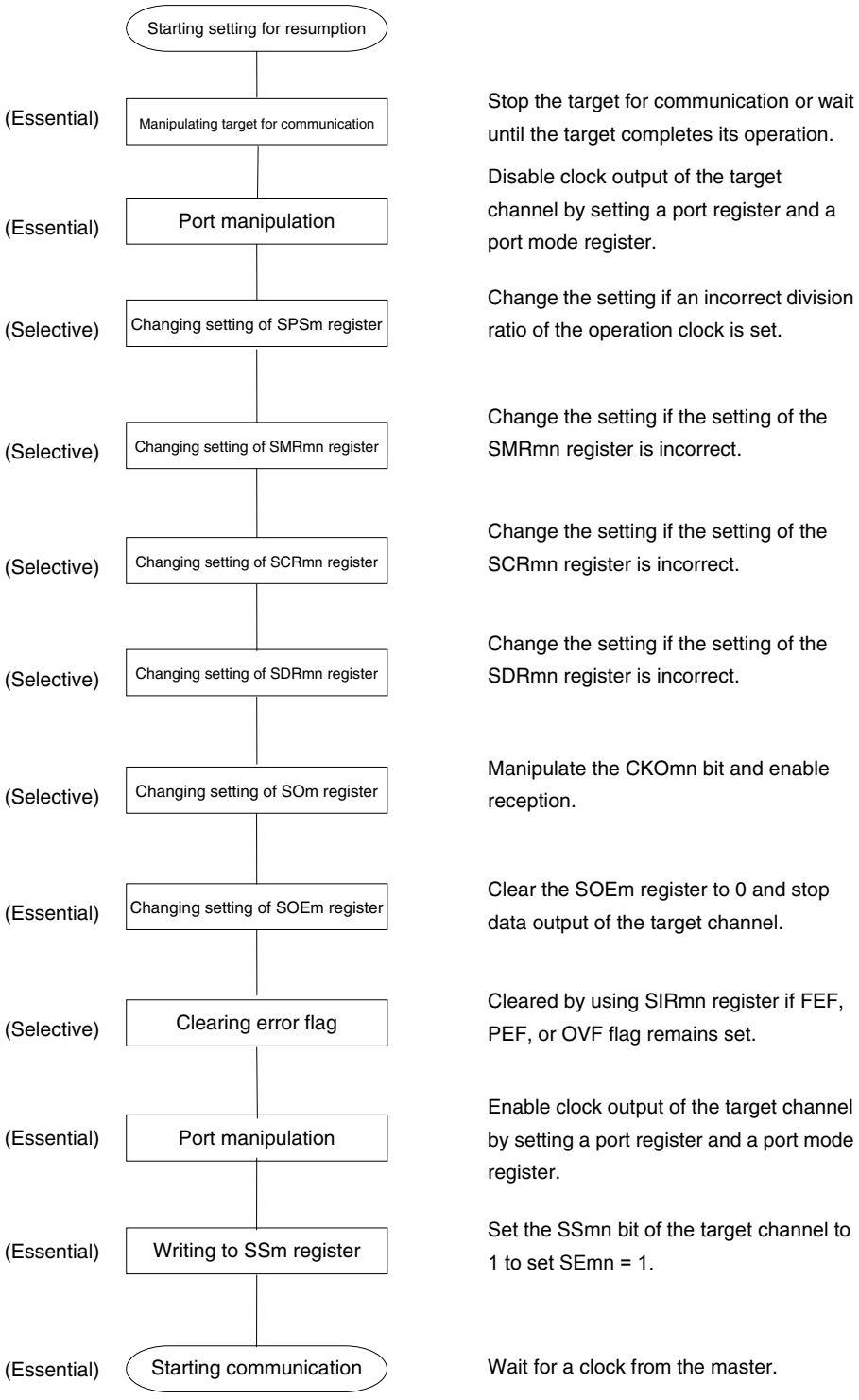
Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

Figure 12-56. Procedure for Stopping Slave Reception



Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

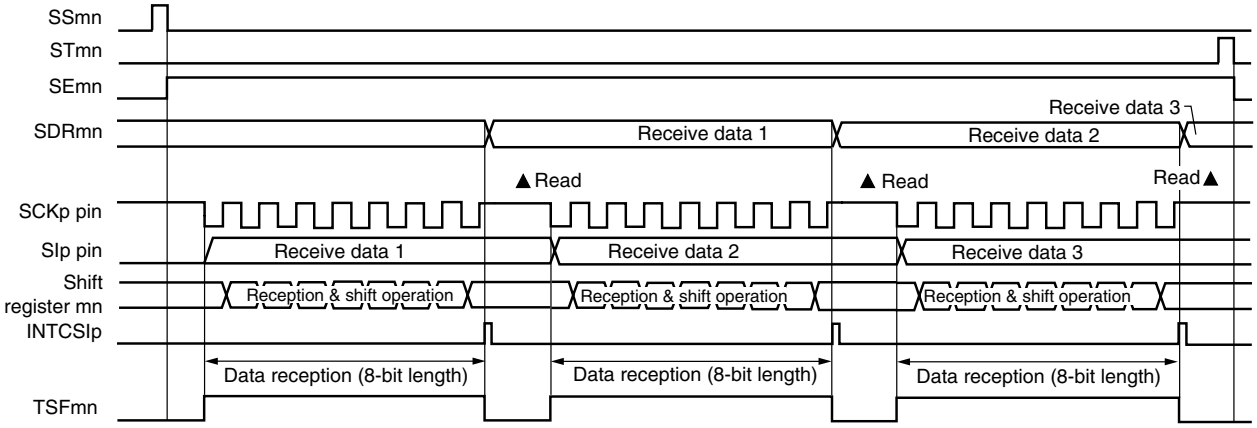
Figure 12-57. Procedure for Resuming Slave Reception



Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

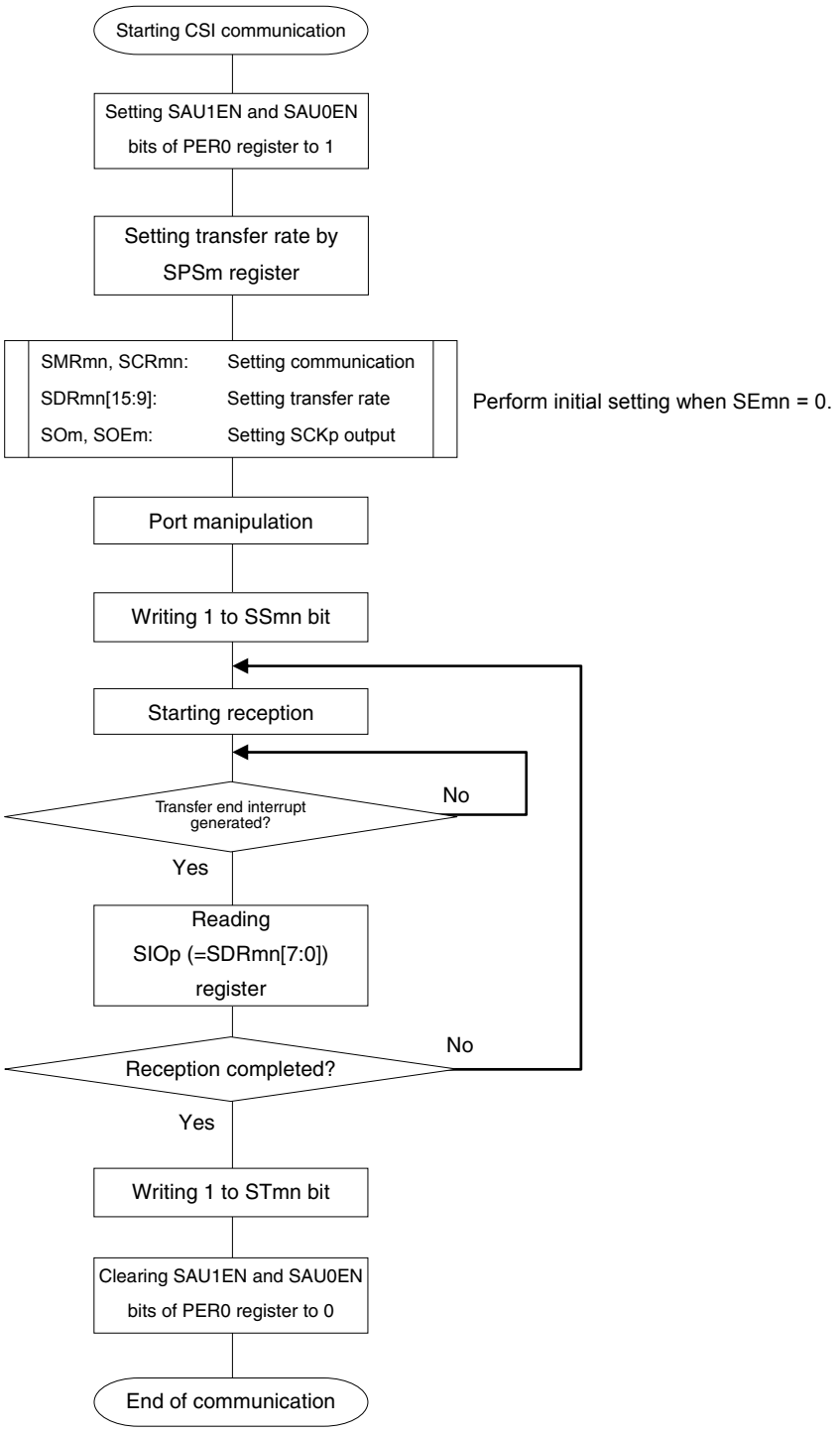
(3) Processing flow (in single-reception mode)

Figure 12-58. Timing Chart of Slave Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

Figure 12-59. Flowchart of Slave Reception (in Single-Reception Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

12.5.6 Slave transmission/reception

Slave transmission/reception is that the 78K0R/Lx3-M microcontrollers transmit/receive data to/from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI20
Target channel	Channel 0 of SAU1
Pins used	$\overline{SCK20}$, SI20, SO20
Interrupt	INTCSI20
	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 [MHz]Notes 1, 2
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data I/O starts from the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Notes 1. Because the external serial clock input to pin $\overline{SCK20}$ is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [MHz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel

2. m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

(1) Register setting

Figure 12-60. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI20)

(a) Serial output register 1 (SO1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10 ×	0	0	0	0	1	SO12 ×	1	SO10 0/1

(b) Serial output enable register 1 (SOE1) … 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
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12 ×	0	SOE10 0/1

(c) Serial channel start register 1 (SS1) … 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
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13 ×	SS12 ×	SS11 ×	SS10 0/1

(d) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10 0/1	CCS10 1	0	0	0	0	0	STS10 0	0	SIS100 0	1	0	0	MD102 0	MD101 0	MD100 0/1

Operation mode of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10 1	RXE10 1	DAP10 0/1	CKP10 0/1	0	EOC10 0	PTC101 0	PTC100 0	DIR10 0/1	0	SLC101 0	SLC100 0	0	DLS102 1	DLS101 1	DLS100 0/1

(f) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

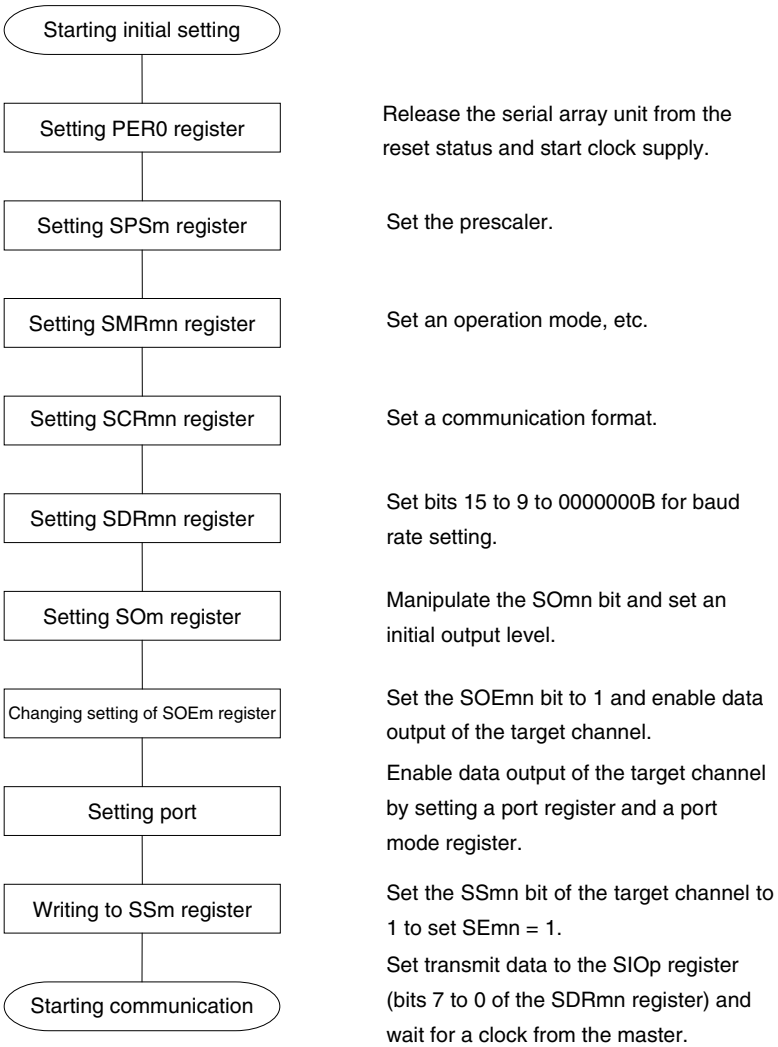
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	0000000 (baud rate setting)							0	Transmit data setting/receive data register							

SIO20

Remark □: 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

(2) Operation procedure

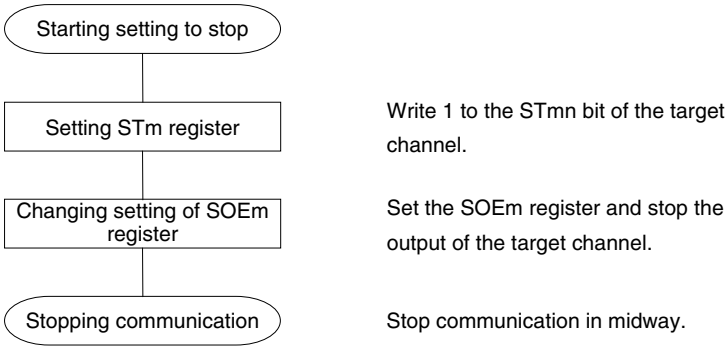
Figure 12-61. Initial Setting Procedure for Slave Transmission/Reception



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

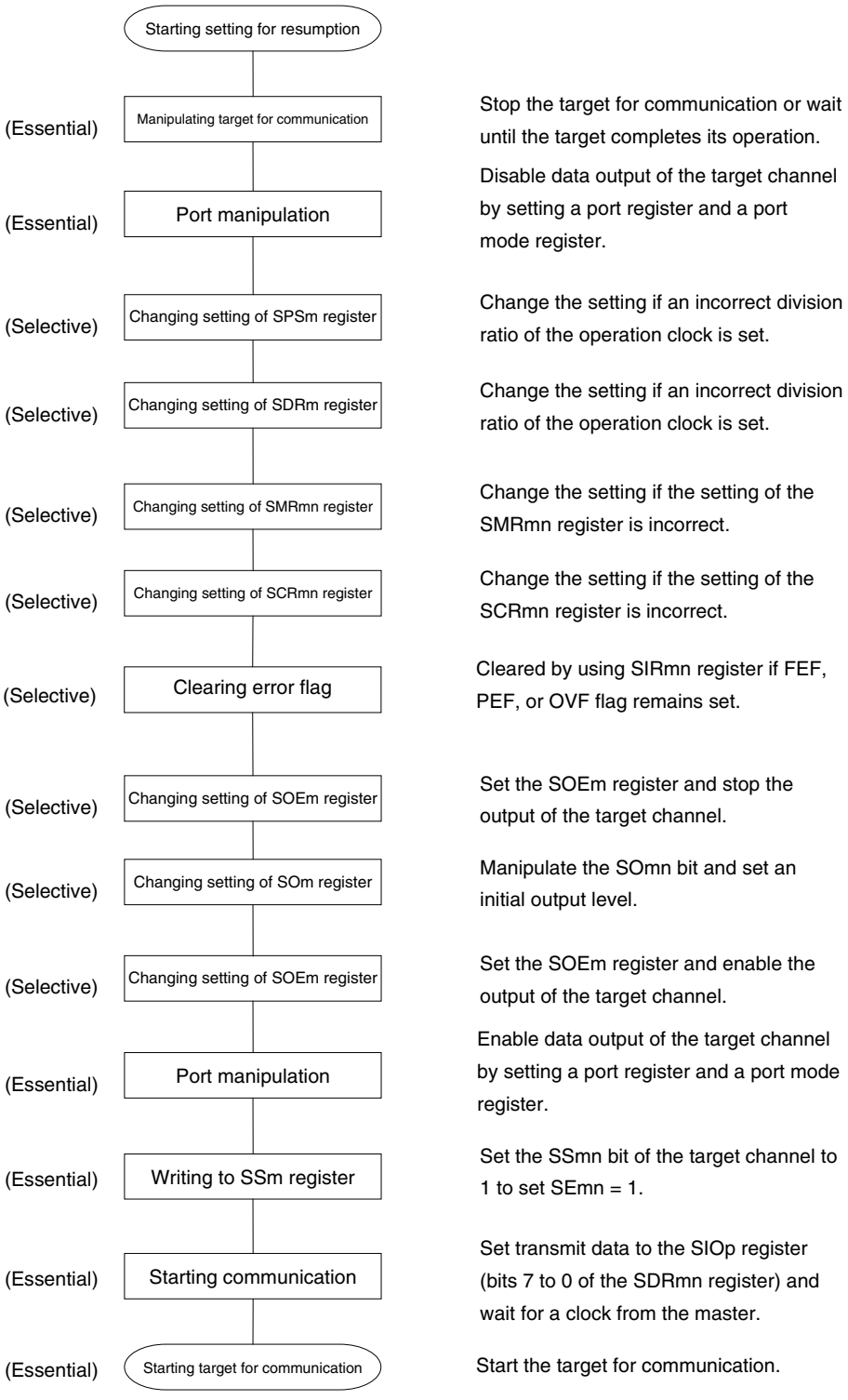
Figure 12-62. Procedure for Stopping Slave Transmission/Reception



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 12-63 Procedure for Resuming Slave Transmission/Reception**).

2. m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

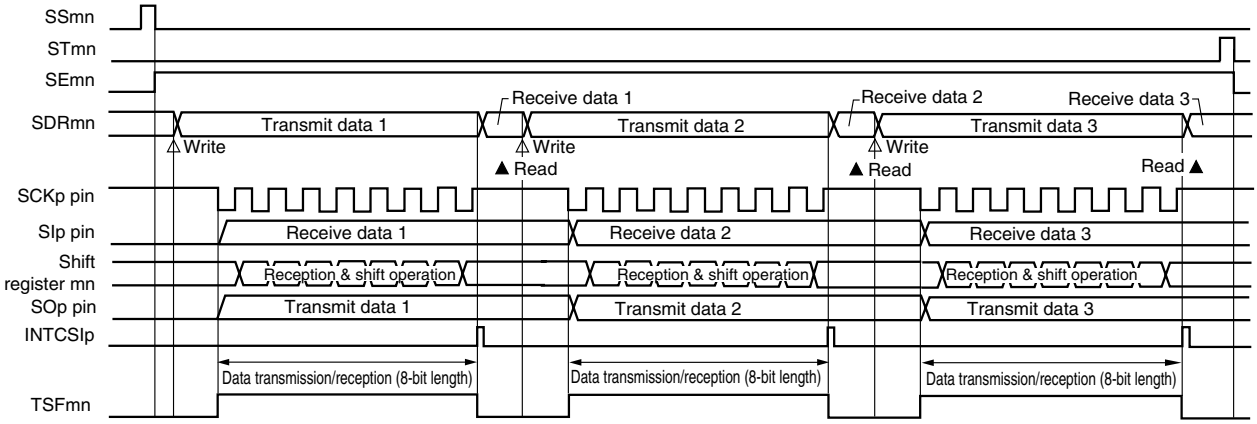
Figure 12-63. Procedure for Resuming Slave Transmission/Reception



Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

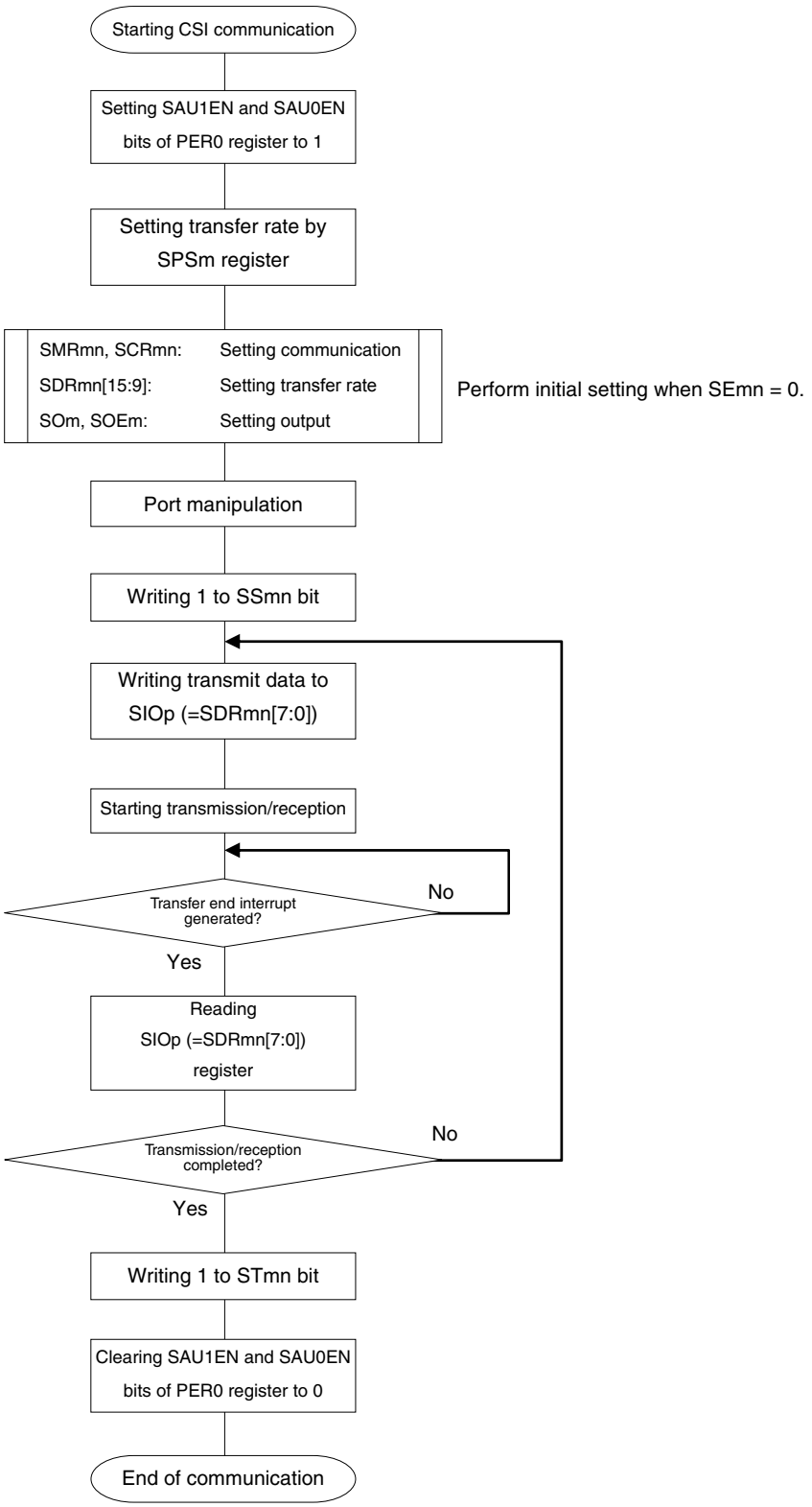
(3) Processing flow (in single-transmission/reception mode)

Figure 12-64. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

Figure 12-65. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)

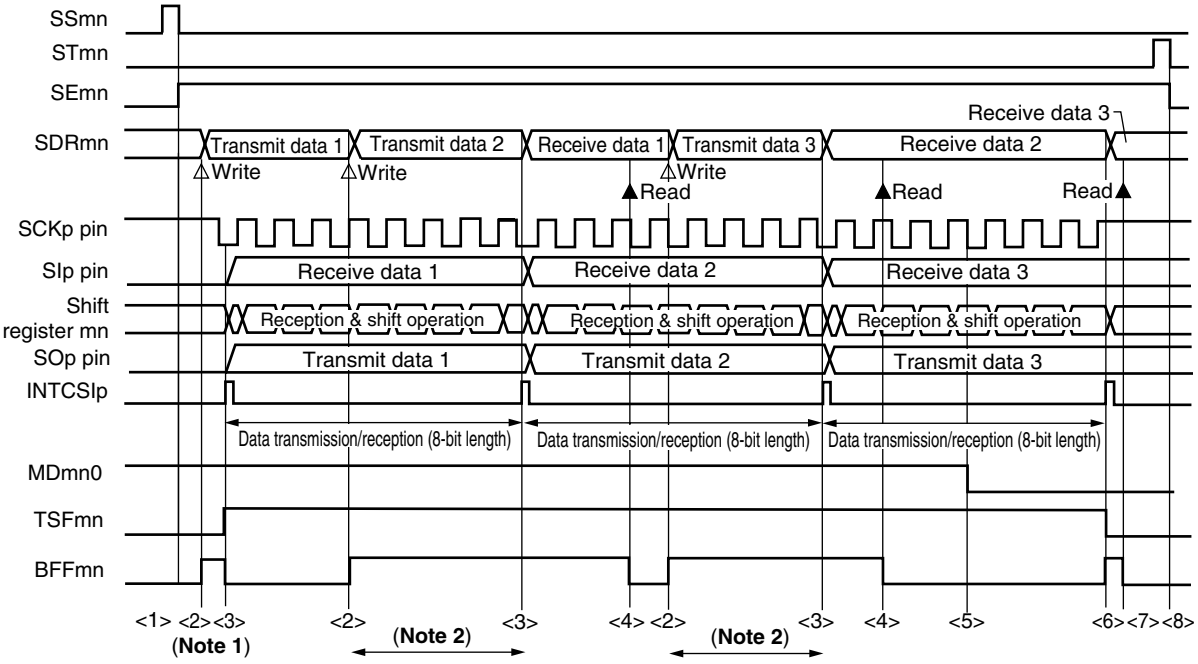


Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

(4) Processing flow (in continuous transmission/reception mode)

Figure 12-66. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 1. When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

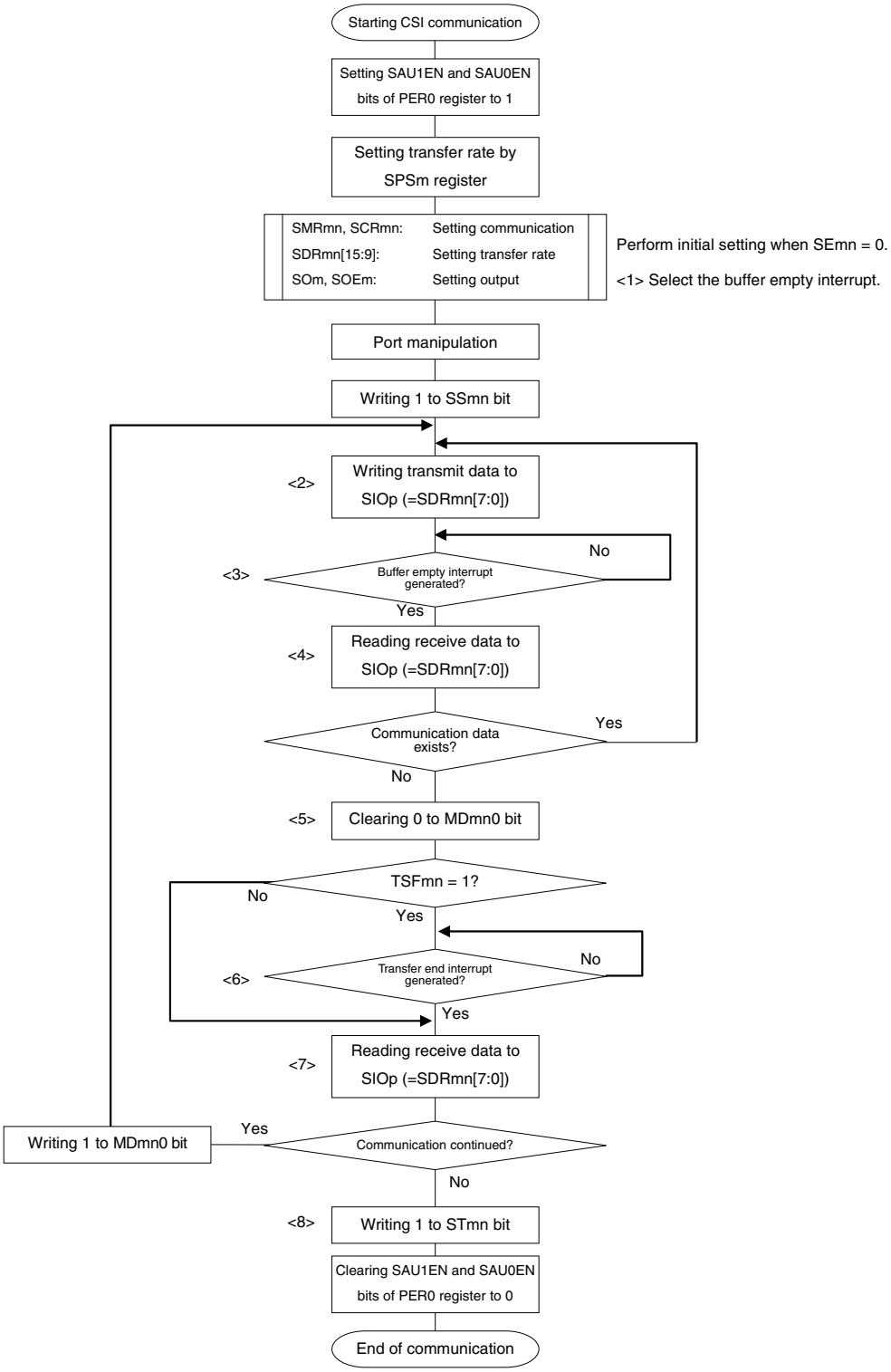
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit 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.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-67 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

Figure 12-67. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-66 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 1), n: Channel number (n = 0), p: CSI number (p = 20), mn = 10

12.5.7 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI20) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (MCK) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock ($\overline{SCK}$) supplied by master}[Note] [Hz]

Note The permissible maximum frequency is the smaller of $f_{CLK}/6$ and $f_{MCK}/2$.

Remarks 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000000B to 1111111B) and therefore is 0 to 127.

2. m: Unit number (m = 1), n: Channel number (n = 0)

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 12-2. Selection of operation clock

SMRmn Register	SPSm Register								Operation Clock (MCK)[Note 1]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	313 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM03 if m = 1[Note 2]	
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	313 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM03 if m = 1[Note 2]	
Other than above									Setting prohibited	

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (STm = 000FH) the operation of the serial array unit (SAUm). When selecting INTTM03 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAUm can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting TIS03 (if m = 1) of the TIS0 register to 1) and selecting INTTM03 by using the SPSm register in channels 3 of TAU0. When changing f_{CLK}, however, SAUm and TAU0 must be stopped as described in Note 1 above.

Remarks 1. X: don't care

2. m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

12.6 Operation of UART (UART0, UART2, UART3) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. It transmits or receives data in asynchronization with the party of communication (by using an internal baud rate). Full-duplex UART communication can be realized by using two channels, one dedicated to transmission (even channel) and the other to reception (odd channel).

[Data transmission/reception]

- Data length of 5, 7, or 8 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

The LIN-bus is supported in UART3 (2, 3 channels of unit 1)

[LIN-bus functions]

- Wakeup signal detection
- Sync break field (SBF) detection
- Sync field measurement, baud rate calculation

} External interrupt (INTP0) or timer array unit (TAU) is used.

UART0 uses channels 0 and 1 of SAU0.
UART2 uses channels 0 and 1 of SAU1.
UART3 uses channels 2 and 3 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	–	UART0	–
	1	–		–
	2	CSI10 (dedicated to the extended SFR (3rd SFR) interface)	–	–
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

Caution When using serial array units 0 and 1 as UARTs, the channels of both the transmitting side (even-number channel) and the receiving side (odd-number channel) can be used only as UARTs.

UART performs the following four types of communication operations.

- UART transmission (See **12.6.1**.)
- UART reception (See **12.6.2**.)
- LIN transmission (UART3 only) (See **12.6.3**.)
- LIN reception (UART 3 only) (See **12.6.4**.)

12.6.1 UART transmission

UART transmission is an operation to transmit data from the 78K0R/Lx3-M microcontrollers to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART0	UART2	UART3
Target channel	Channel 0 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	TxD0	TxD2	TxD3
Interrupt	INTST0	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	5, 7, or 8 bits		
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps]Note		
Data phase	Forward output (default: high level) Reverse output (default: low level)		
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity		
Stop bit	The following selectable • Appending 1 bit • Appending 2 bits		
Data direction	MSB or LSB first		

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 12-68. Example of Contents of Registers for UART Transmission of UART (UART0, UART2, UART3) (1/2)

(a) Serial output register m (SOm) … 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
SOm	0	0	0	0	1	CKO02 Note 1	CKO01 ×	CKO10 ×	0	0	0	0	1	SO12 0/1 Note 2	1	SOm0 0/1 Note 2

(b) 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	SOE12 0/1	0	SOEm0 0/1

(c) 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	SS13 ×	SS12 0/1	SSm1 ×	SSm0 0/1

(d) Serial output level register m (SOLm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL12 0/1	0	SOLm0 0/1

0: Forward (normal) transmission
1: Reverse transmission

(e) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0/1

Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

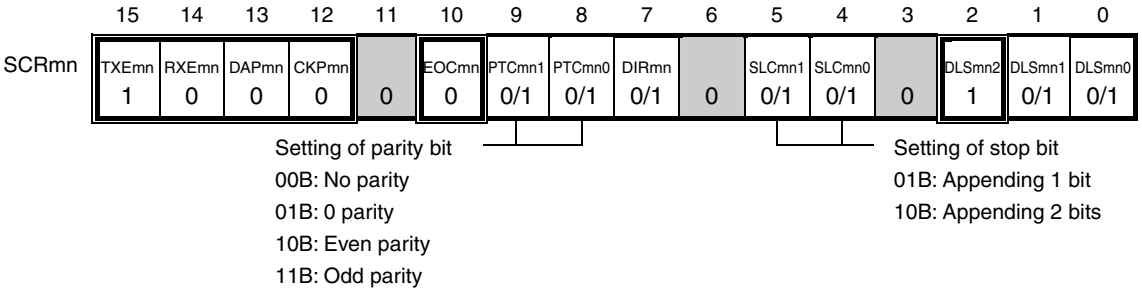
Notes 1. Bit can be used in extended SFR (3rd SFR) interface only.
See **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE.**

2. 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.

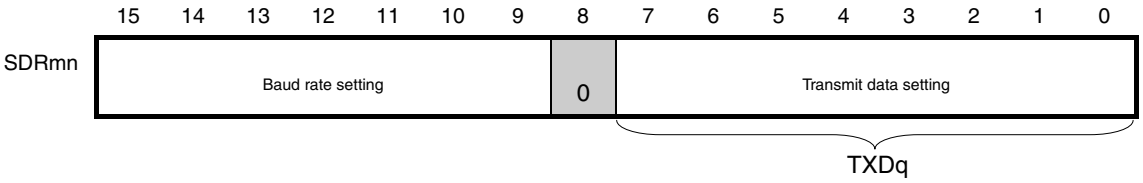
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 10, 12
□: Setting is fixed in the UART transmission mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-68. Example of Contents of Registers for UART Transmission of UART (UART0, UART2, UART3) (2/2)

(f) Serial communication operation setting register mn (SCRmn)



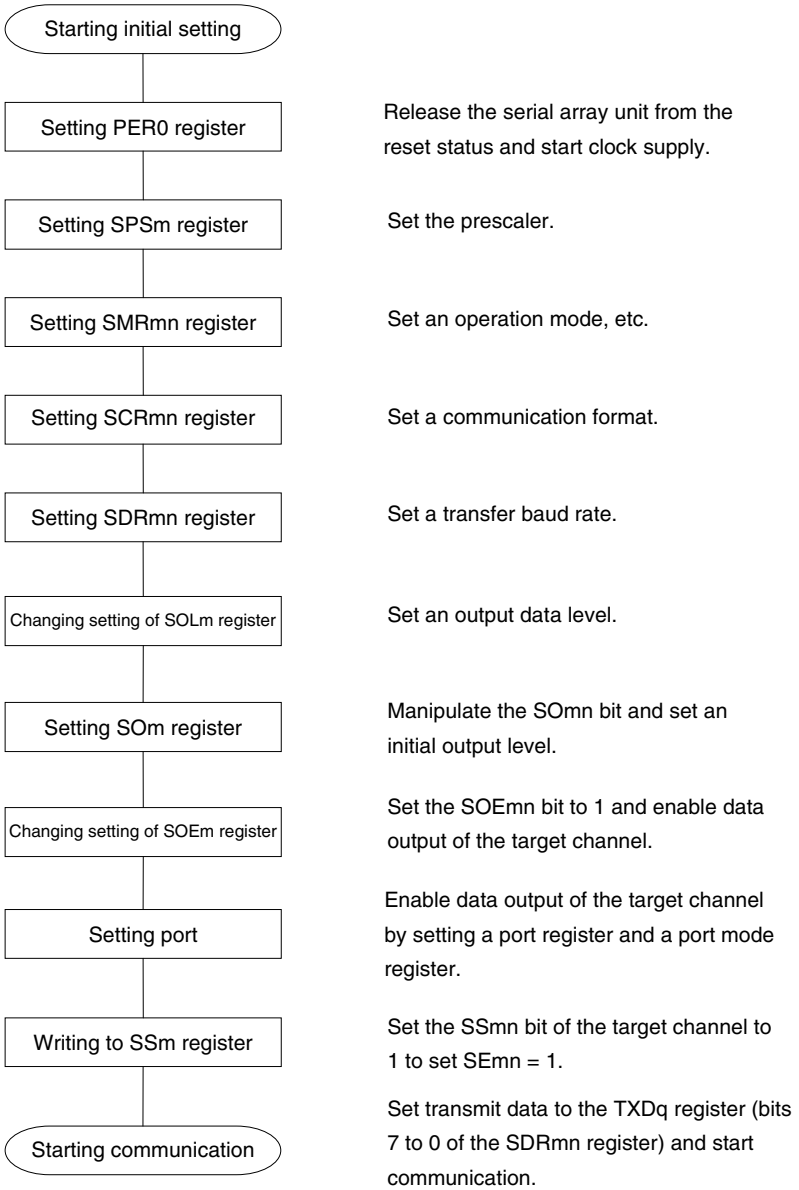
(g) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0, 2, 3), mn = 00, 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

(2) Operation procedure

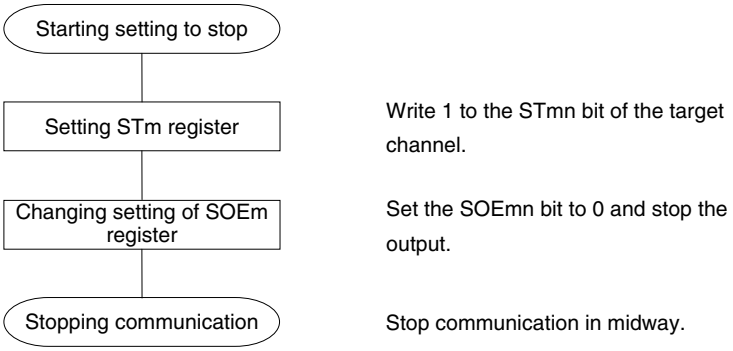
Figure 12-69. Initial Setting Procedure for UART Transmission



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0, 2, 3), mn = 00, 10, 12

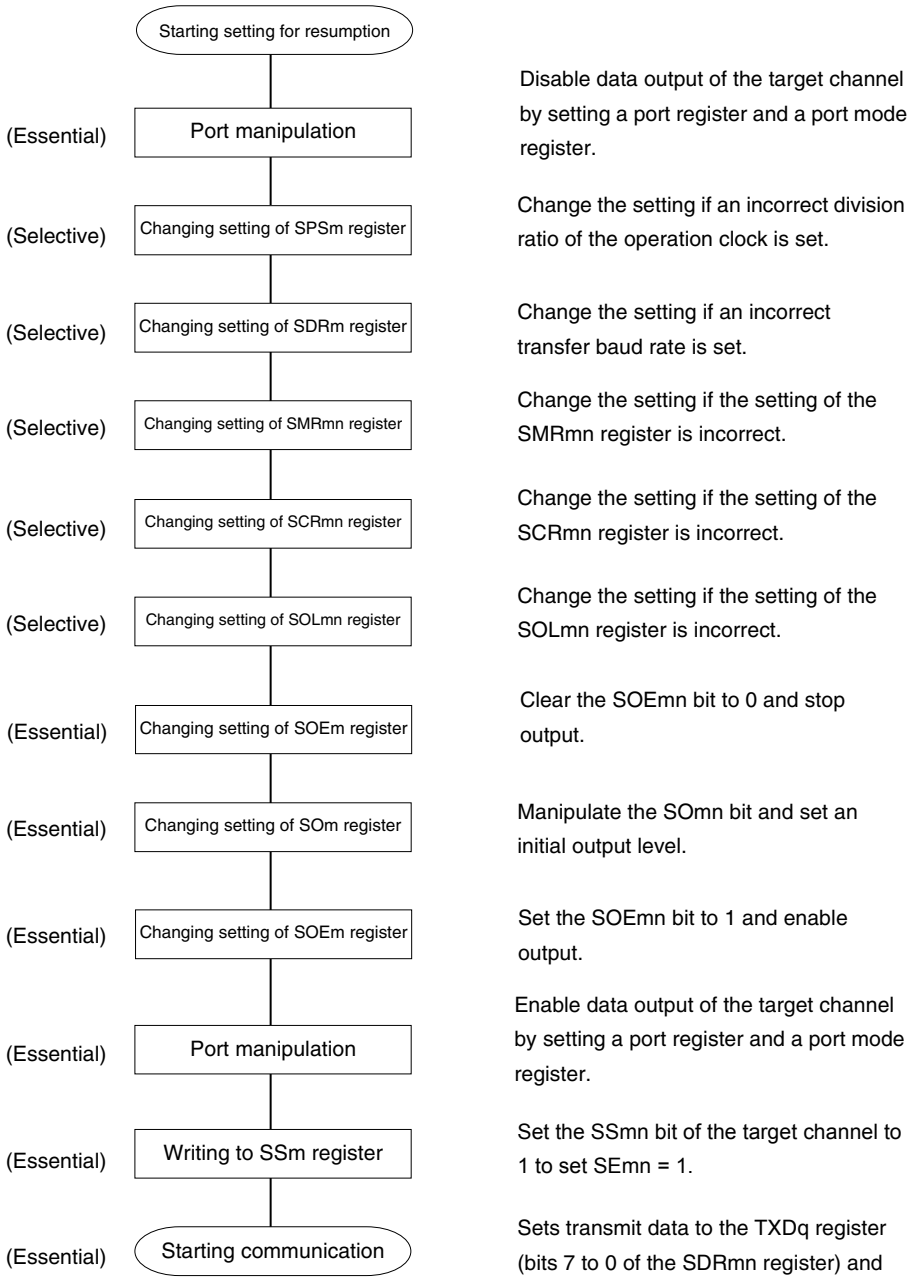
Figure 12-70. Procedure for Stopping UART Transmission



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 12-71 Procedure for Resuming UART Transmission**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 10, 12

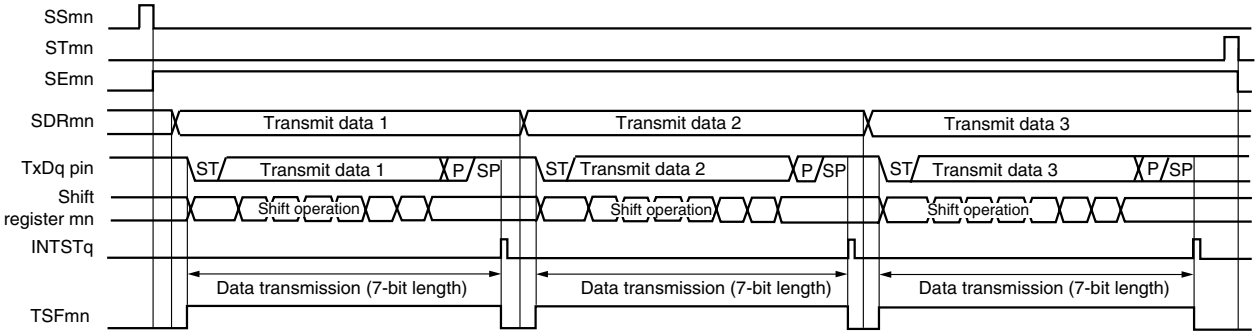
Figure 12-71. Procedure for Resuming UART Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0, 2, 3), mn = 00, 10, 12

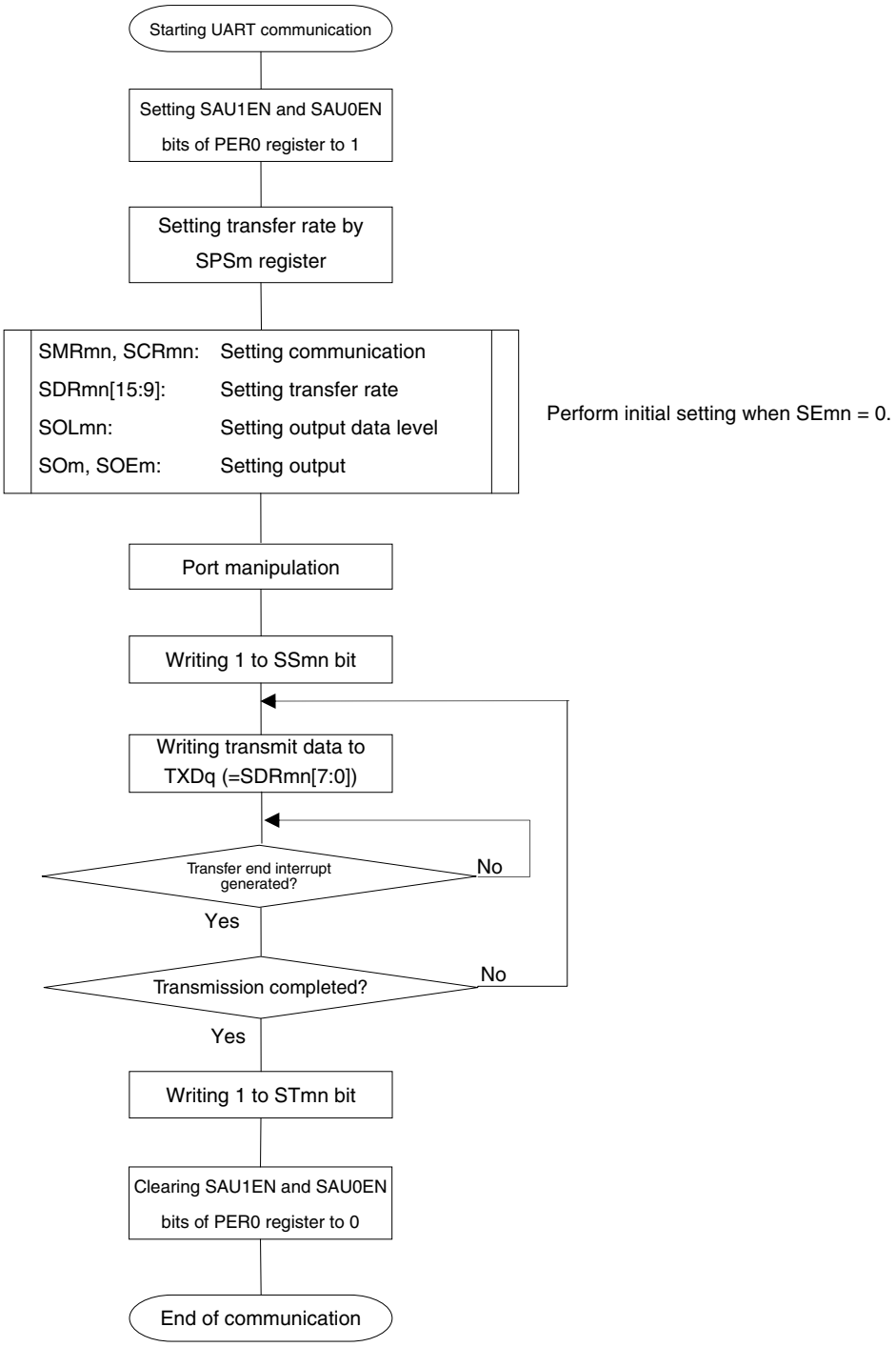
(3) Processing flow (in single-transmission mode)

Figure 12-72. 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, 2, 3), mn = 00, 10, 12

Figure 12-73. Flowchart of UART Transmission (in Single-Transmission Mode)

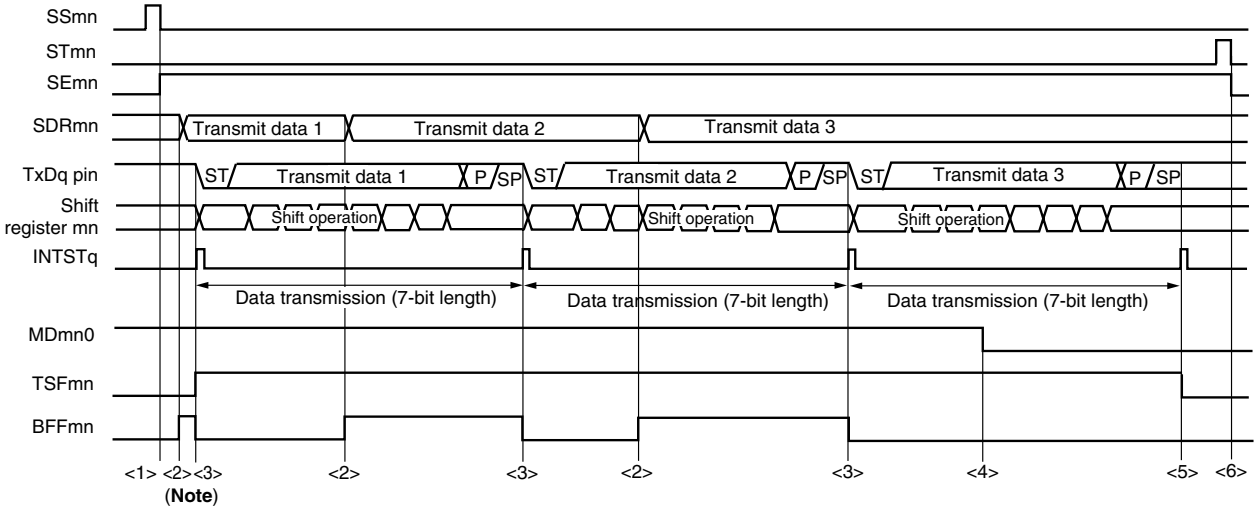


Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0, 2, 3), mn = 00, 10, 12

(4) Processing flow (in continuous transmission mode)

Figure 12-74. Timing Chart of UART Transmission (in Continuous Transmission Mode)

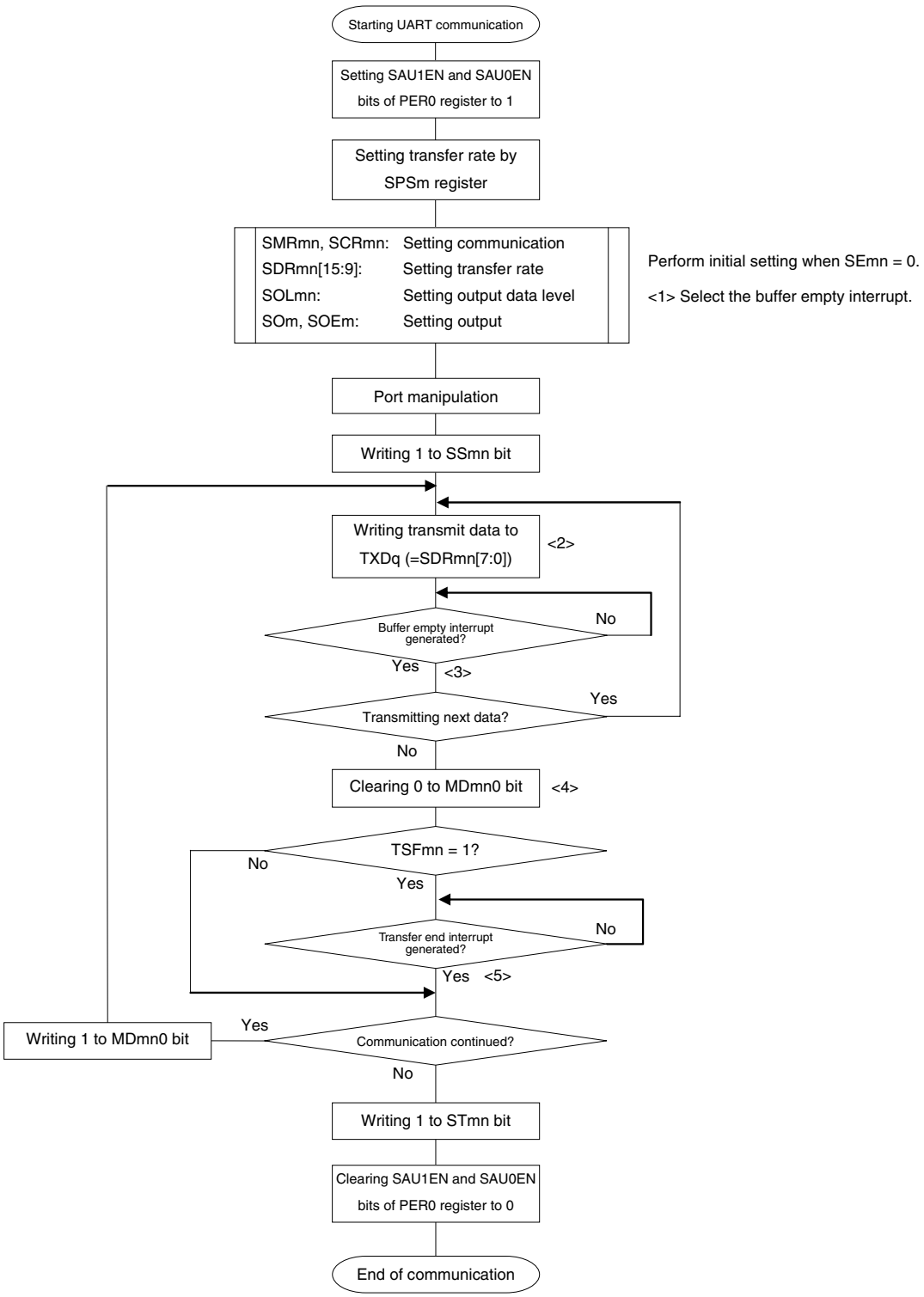


Note When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

Caution The MDmn0 bit 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.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0, 2, 3), mn = 00, 10, 12

Figure 12-75. Flowchart of UART Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. <1> to <5> in the figure correspond to <1> to <5> in **Figure 12-74 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0, 2, 3), mn = 00, 10, 12

12.6.2 UART reception

UART reception is an operation wherein the 78K0R/Lx3-M microcontrollers asynchronously receive 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	UART2	UART3
Target channel	Channel 1 of SAU0	Channel 1 of SAU1	Channel 3 of SAU1
Pins used	RxD0	RxD2	RxD3
Interrupt	INTSR0	INTSR2	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error interrupt	INTSRE0	INTSRE2	INTSRE3
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)		
Transfer data length	5, 7 or 8 bits		
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps]Note		
Data phase	Forward output (default: high level) Reverse output (default: low level)		
Parity bit	The following selectable • No parity bit (no parity check) • Appending 0 parity (no parity check) • Appending even parity • Appending odd parity		
Stop bit	Appending 1 bit		
Data direction	MSB or LSB first		

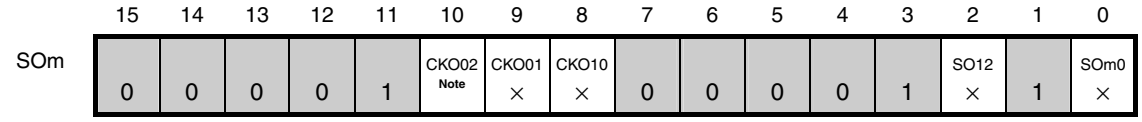
Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

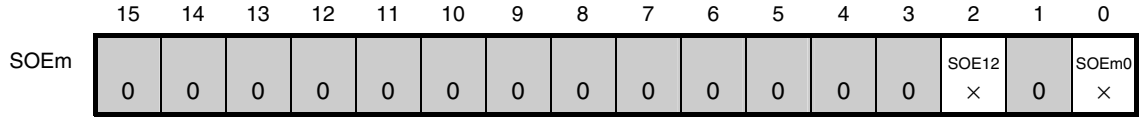
(1) Register setting

Figure 12-76. Example of Contents of Registers for UART Reception of UART (UART0, UART2, UART3)

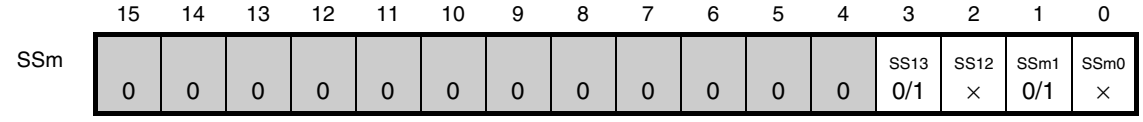
(a) Serial output register m (SOm) … Unused in this mode.



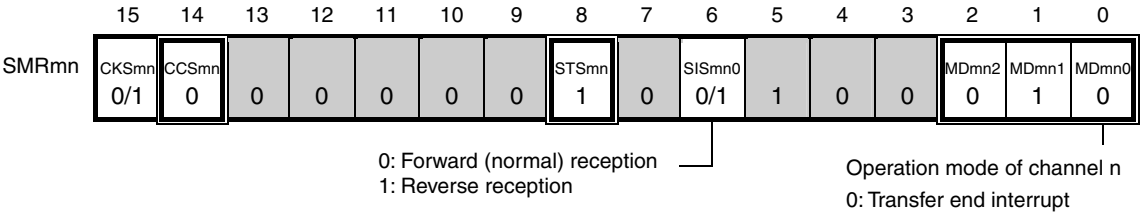
(b) Serial output enable register m (SOEm) … Unused in this mode.



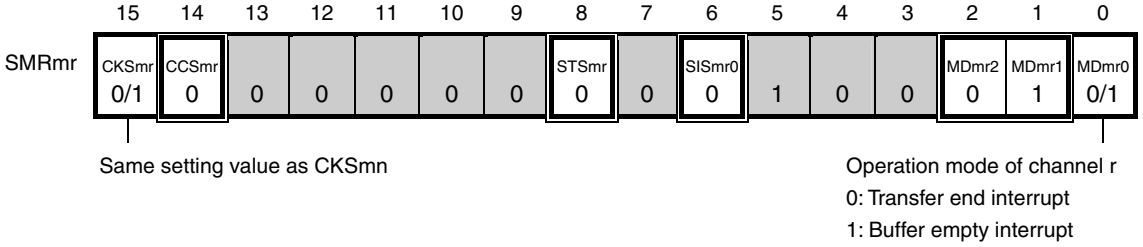
(c) Serial channel start register m (SSm) … Sets only the bits of the target channel is 1.



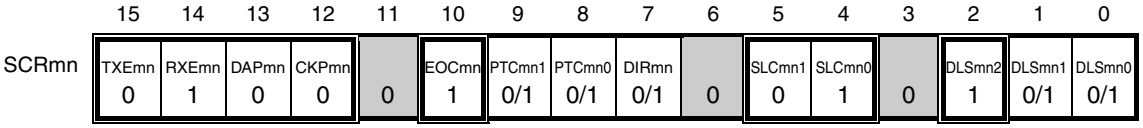
(d) Serial mode register mn (SMRmn)



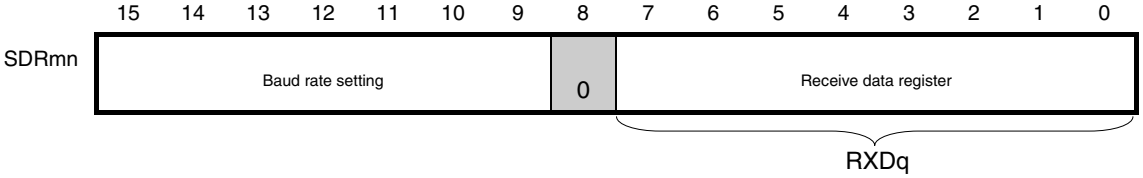
(e) Serial mode register mr (SMRmr)



(f) Serial communication operation setting register mn (SCRmn)



(g) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



Note Bit can be used in extended SFR (3rd SFR) interface only.
See **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE.**

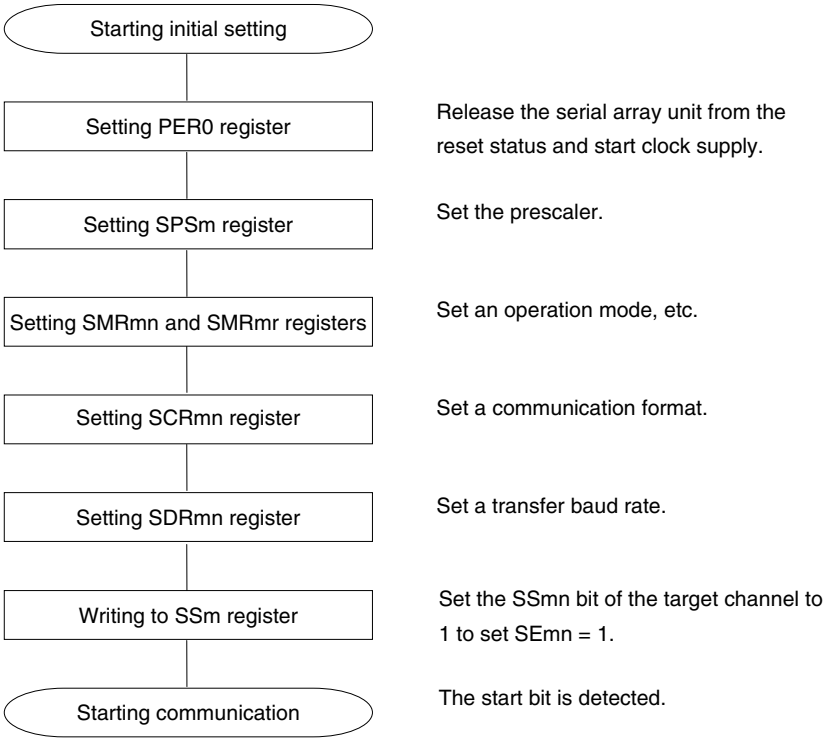
(**Caution** and **Remark** are listed on the next page.)

Caution For the UART reception, be sure to set SMRmr of channel r that is to be paired with channel n.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), r: Channel number (r = n – 1),
q: UART number (q = 0, 2, 3), mn = 01, 11, 13, mr = 00, 10, 12
□: Setting is fixed in the UART reception mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

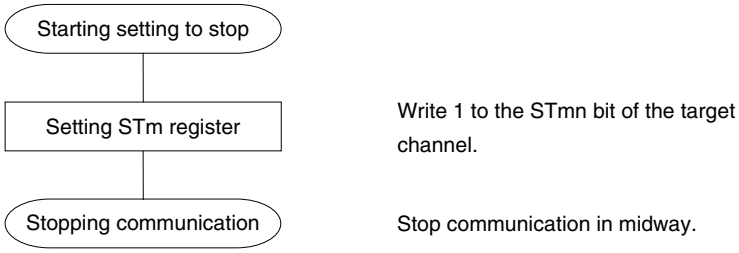
Figure 12-77. Initial Setting Procedure for UART Reception



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

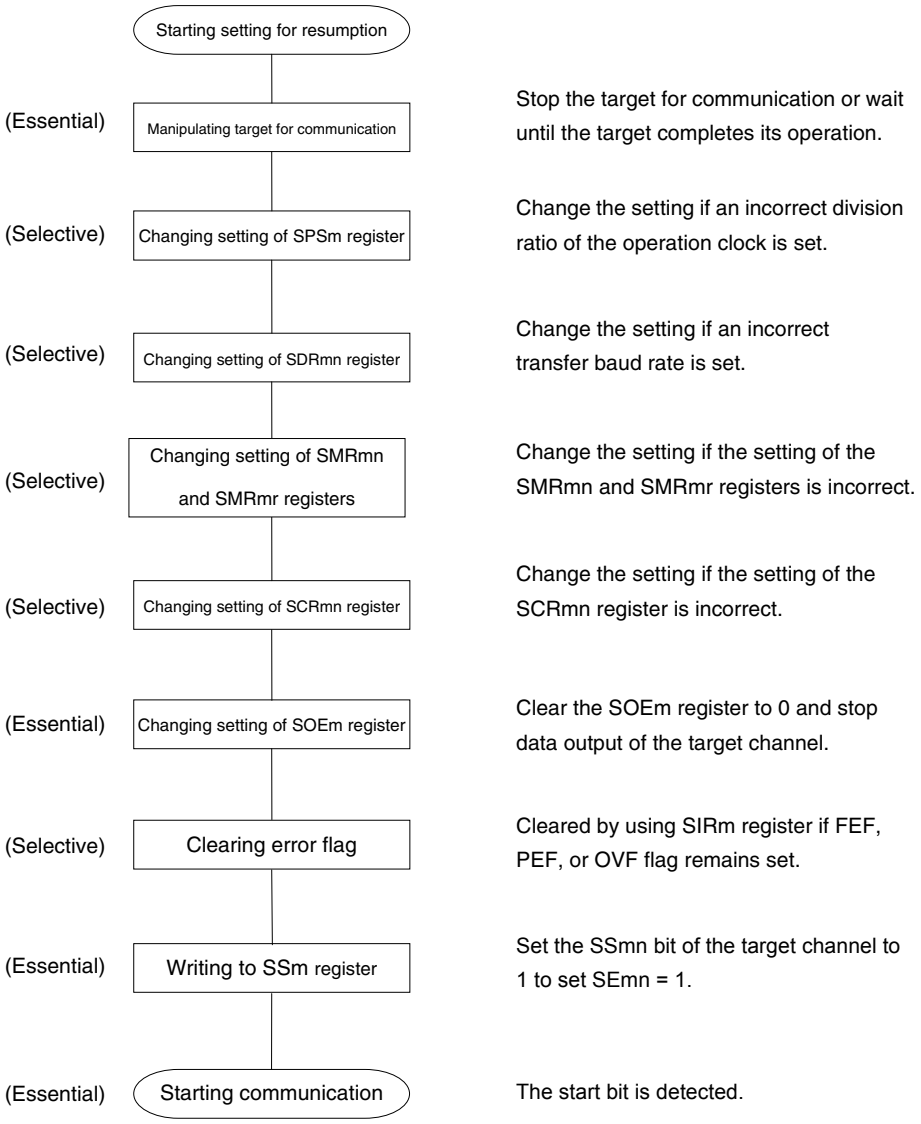
Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), r: Channel number (r = n – 1), mn = 01, 11, 13, mr = 00, 10, 12

Figure 12-78. Procedure for Stopping UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 11, 13

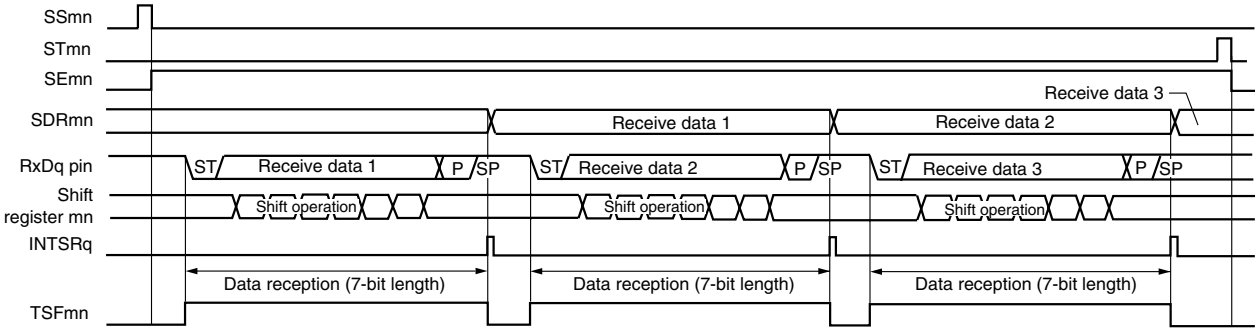
Figure 12-79. Procedure for Resuming UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), r: Channel number (r = n – 1), mn = 01, 11, 13, mr = 00, 10, 12

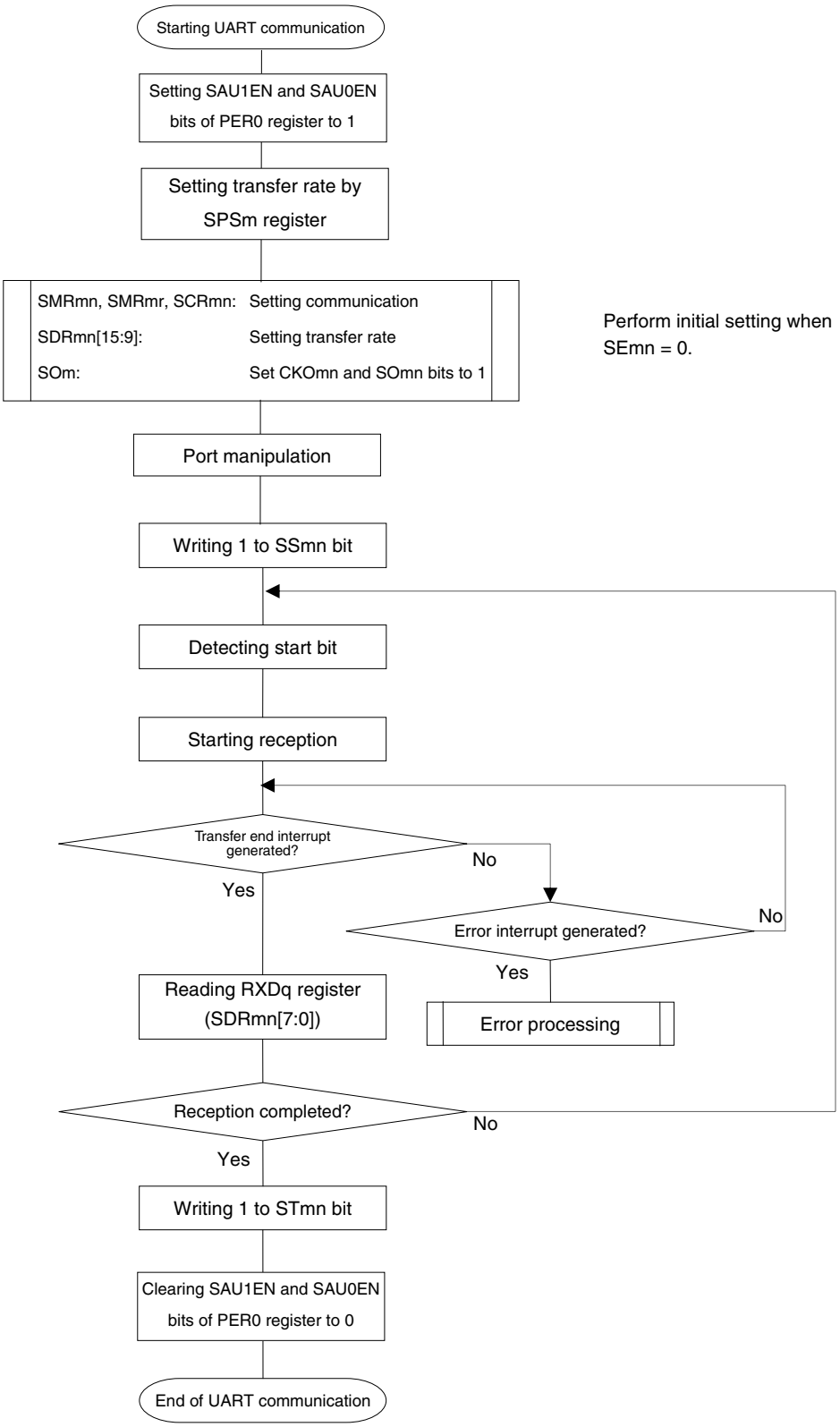
(3) Processing flow

Figure 12-80. Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), q: UART number (q = 0, 2, 3), mn = 01, 11, 13

Figure 12-81. Flowchart of UART Reception



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), r: Channel number (r = n – 1),
q: UART number (q = 0, 2, 3), mn = 01, 11, 13, mr = 00, 10, 12

12.6.3 LIN transmission

Of UART transmission, UART3 supports LIN communication.

For LIN transmission, channel 2 of unit 1 (SAU1) is used.

UART	UART0	UART2	UART3
Support of LIN communication	Not supported	Not supported	Supported
Target channel	–	–	Channel 2 of SAU1
Pins used	–	–	TxD3
Interrupt	–	–	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	8 bits		
Transfer rate	Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps]Note		
Data phase	Forward output (default: high level) Reverse output (default: low level)		
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity		
Stop bit	The following selectable • Appending 1 bit • Appending 2 bits		
Data direction	MSB or LSB first		

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) 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.

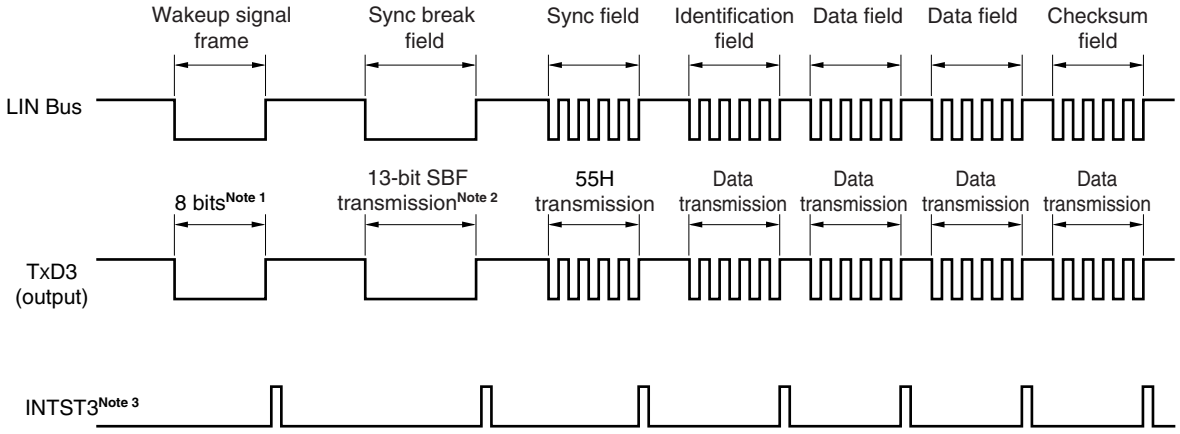
Usually, the master is connected to a network such as CAN (Controller Area Network).

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 12-82 outlines a transmission operation of LIN.

Figure 12-82. Transmission Operation of LIN



Notes 1. The baud rate is set so as to satisfy the standard of the wakeup signal and data of 00H is transmitted.

2. A sync 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 sync break field is calculated as follows.

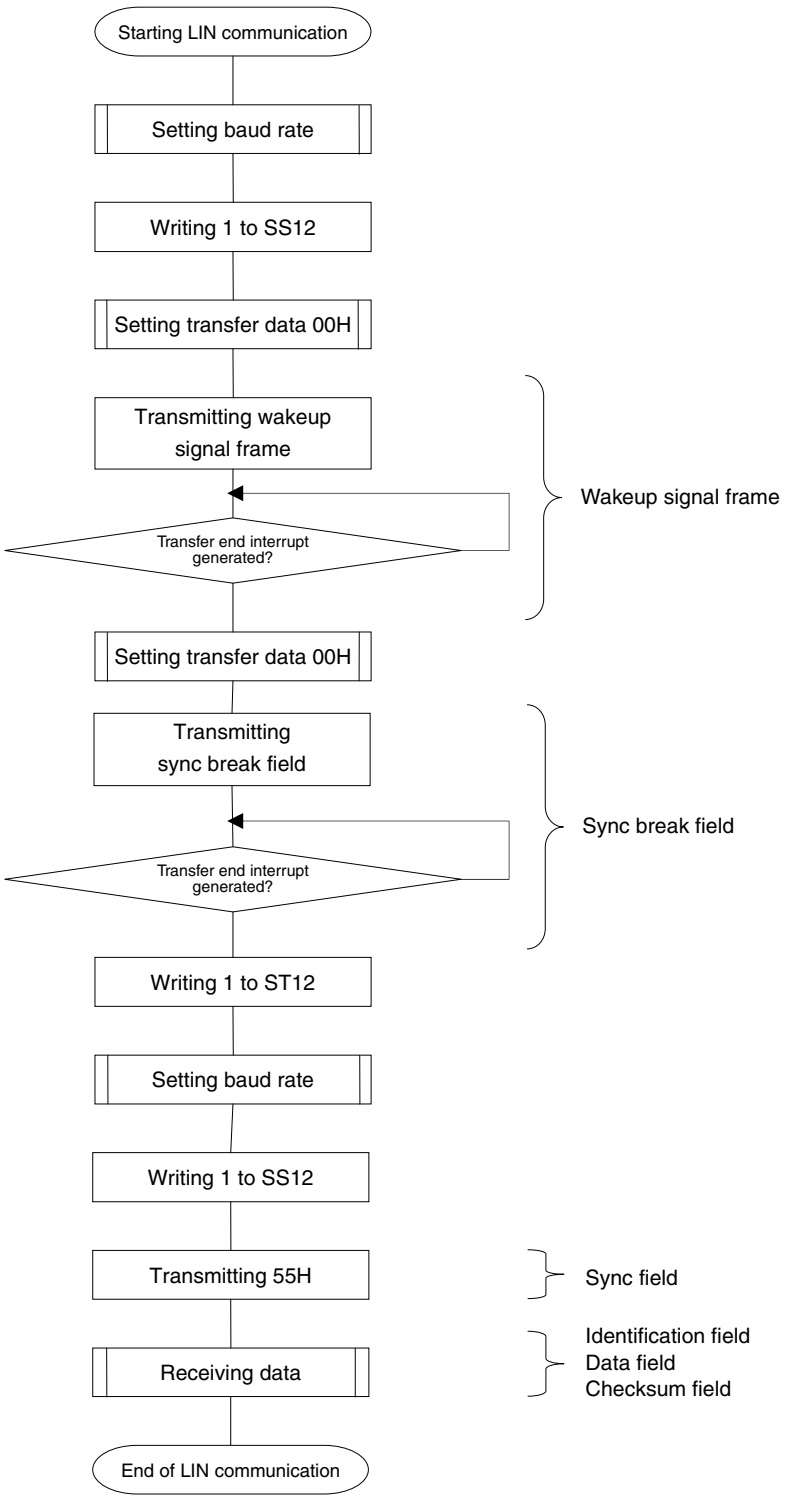
(Baud rate of sync break field) = 9/13 × N

By transmitting data of 00H at this baud rate, a sync break field is generated.

3. INTST3 is output upon completion of transmission. INTST3 is also output when SBF transmission is executed.

Remark The interval between fields is controlled by software.

Figure 12-83. Flowchart for LIN Transmission



12.6.4 LIN reception

Of UART reception, UART3 supports LIN communication.

For LIN reception, channel 3 of unit 1 (SAU1) is used.

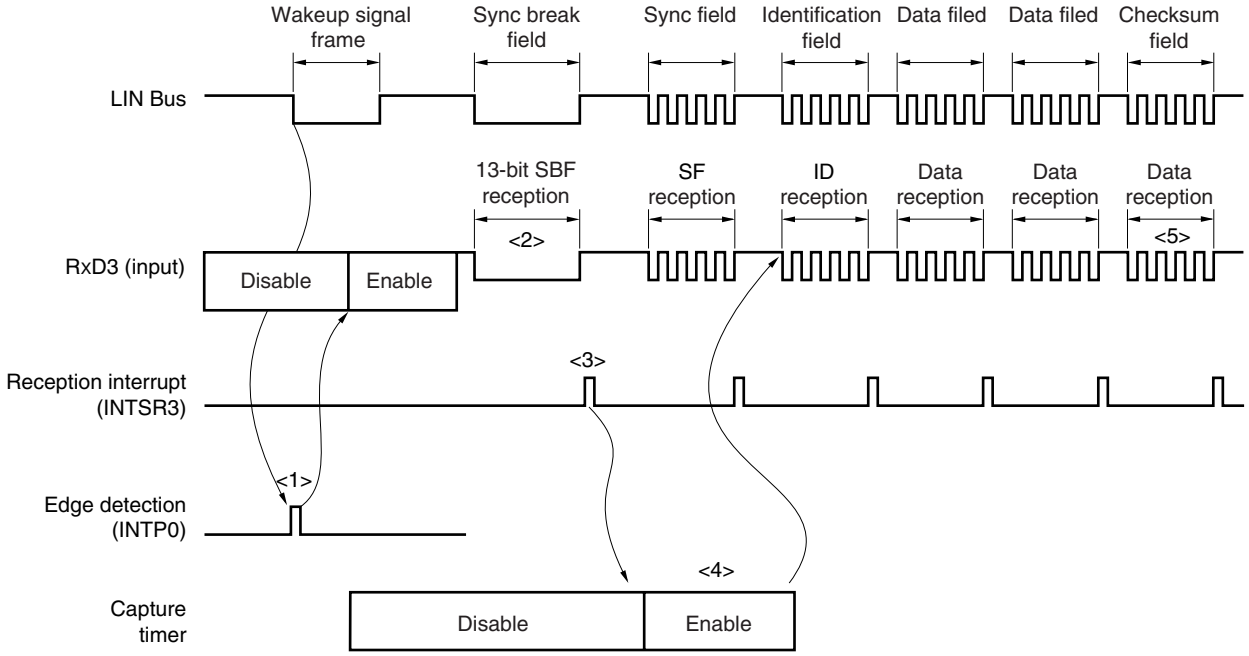
UART	UART0	UART2	UART3
Support of LIN communication	Not supported	Not supported	Supported
Target channel	–	–	Channel 3 of SAU1
Pins used	–	–	RxD3
Interrupt	–	–	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error interrupt	–	–	INTSRE3
Error detection flag	• Framing error detection flag (FEF13) • Parity error detection flag (PEF13) • Overrun error detection flag (OVF13)		
Transfer data length	8 bits		
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps]Note		
Data phase	Forward output (default: high level) Reverse output (default: low level)		
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity		
Stop bit	The following selectable • Appending 1 bit • Appending 2 bits		
Data direction	MSB or LSB first		

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

Figure 12-84 outlines a reception operation of LIN.

Figure 12-84. 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, enable reception of UART3 (RXE13 = 1) and wait for SBF reception.

<2> When the start bit of SBF is detected, reception is started and serial data is sequentially stored in the RXD3 register (= bits 7 to 0 of the serial data register 13 (SDR13)) at the set baud rate. When the stop bit is detected, the reception end interrupt request (INTSR3) is generated. When data of low levels of 11 bits or more is detected as SBF, it is judged that SBF reception has been correctly completed. If data of low levels of less than 11 bits is detected as SBF, it is judged that an SBF reception error has occurred, and the system returns to the SBF reception wait status.

<3> When SBF reception has been correctly completed, start channel 7 of the timer array unit and measure the bit interval (pulse width) of the sync field (see **6.7.5 Operation as input signal high-/low-level width measurement**).

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART3 once and adjust (re-set) the baud rate.

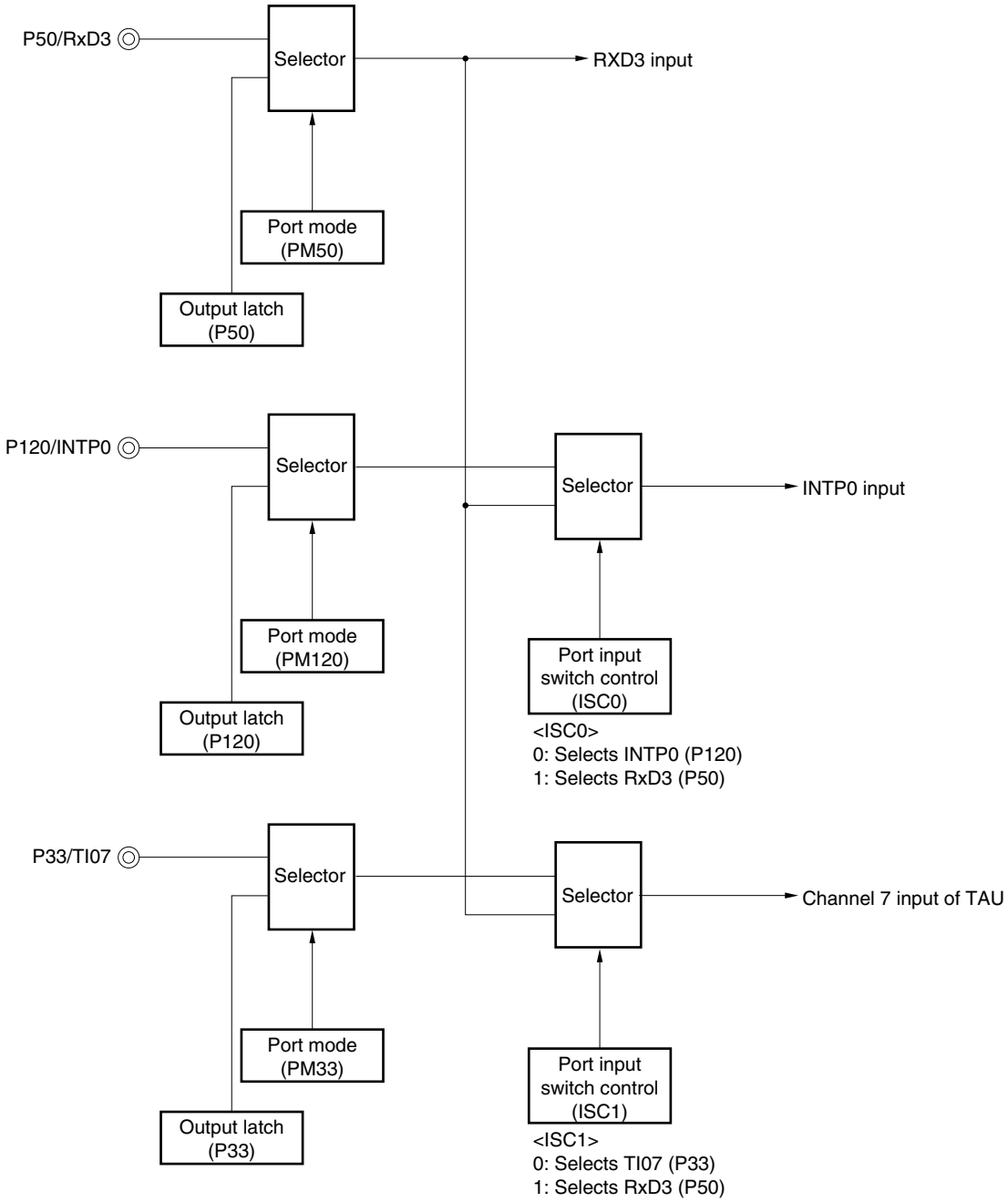
<5> The checksum field should be distinguished by software. In addition, processing to initialize UART3 after the checksum field is received and to wait for reception of SBF should also be performed by software.

Figure 12-85 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 (TAU) to calculate a baud-rate error.

By controlling switch of port input (ISC0/ISC1), the input source of port input (RxD3) for reception can be input to the external interrupt pin (INTP0) and timer array unit (TAU).

Figure 12-85. Port Configuration for Manipulating Reception of LIN



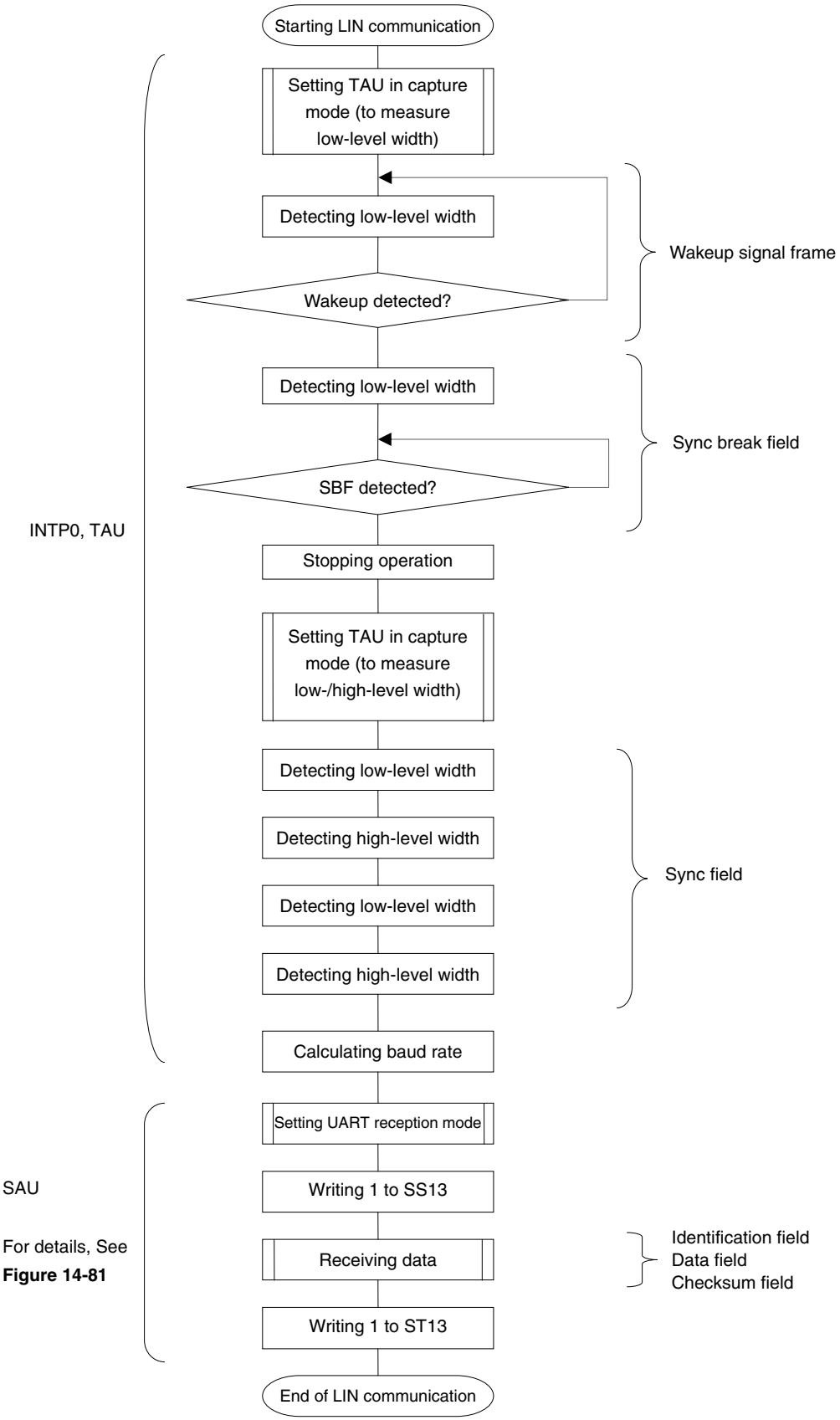
Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 12-17**.)

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 (TAU); Baud rate error 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 RxD3 is measured in the capture mode.)
- Channels 2 and 3 (UART3) of serial array unit 1 (SAU1)

Figure 12-86. Flowchart of LIN Reception



12.6.5 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0, UART2, UART3) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (MCK) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting 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, 01, 10 to 13

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 12-3. Selection of operation clock

SMRmn Register	SPSm Register								Operation Clock (MCK)[Note 1]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	313 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0[Note 2] INTTM03 if m = 1	
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	313 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0[Note 2] INTTM03 if m = 1	
Other than above									Setting prohibited	

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (STm = 000FH) the operation of the serial array unit (SAUm). When selecting INTTM03 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAUm can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting TIS03 (if m = 1) of the TIS0 register to 1) and selecting INTTM03 by using the SPSm register in channels 3 of TAU0. When changing f_{CLK}, however, SAUm and TAU0 must be stopped as described in Note 1 above.

Remarks 1. X: don't care

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

(2) Baud rate error during transmission

The baud rate error of UART (UART0, UART2, 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.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 − 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 20 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 20 MHz			
	Operation Clock (MCK)	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	64	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^8$	64	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^7$	64	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^6$	64	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^5$	64	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^4$	64	9615.38 bps	+0.16 %
19200 bps	$f_{CLK}/2^3$	64	19230.8 bps	+0.16 %
31250 bps	$f_{CLK}/2^3$	39	31250.0 bps	±0.0 %
38400 bps	$f_{CLK}/2^2$	64	38461.5 bps	+0.16 %
76800 bps	$f_{CLK}/2$	64	76923.1 bps	+0.16 %
153600 bps	f_{CLK}	64	153846 bps	+0.16 %
312500 bps	f_{CLK}	31	312500 bps	±0.0 %

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0, UART2, 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.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

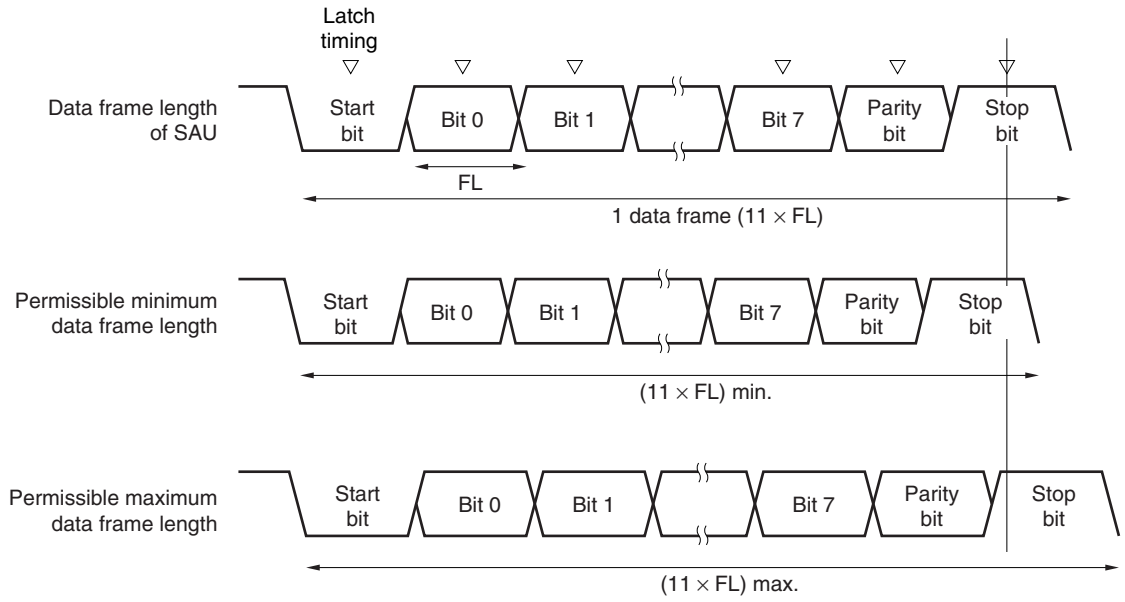
$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **12.6.5 (1) Baud rate calculation expression**.)

k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Figure 12-87. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 12-87, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of the 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.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

12.7 Operation of Simplified I²C (IIC10, IIC20) 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 and does not have a wait detection function. Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[Note] and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Parity error (ACK error)

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection function

Note An ACK is not output when the last data is being received by writing 0 to the SOEmn (SOEm register) bit and stopping the output of serial communication data. See **12.7.3 (2) Processing flow** for details.

Remarks 1. To use the full-function I²C bus, see **CHAPTER 15 SERIAL INTERFACE IICA**.

2. m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

The channels supporting simplified I²C (IIC20) is channel 0 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C
0	0	–	UART0	–
	1	–		–
	2	CSI10 (dedicated to the extended SFR (3rd SFR) interface)	–	–
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN–bus)	–
	3	–		–

Simplified I^2C (IIC20) performs the following four types of communication operations.

- Address field transmission (See **12.7.1**.)
- Data transmission (See **12.7.2**.)
- Data reception (See **12.7.3**.)
- Stop condition generation (See **12.7.4**.)

12.7.1 Address field transmission

Address field transmission is a transmission operation that first executes in I^2C 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^2C	IIC20
Target channel	Channel 0 of SAU1
Pins used	SCL20, SDA20[Note]
Interrupt	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Parity error detection flag (PEFmn)
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)
Transfer rate	Max. f_{CLK}/4 [MHz] (SDRmn [15:9] = 1 or more) f_{CLK}: System clock frequency However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)
Data level	Forward output (default: high level)
Parity bit	No parity bit
Stop bit	Appending 1 bit (for ACK reception timing)
Data direction	MSB first

Note To perform communication via simplified I^2C, set the data I/O pin (SDA20) in the N-ch open-drain output (V_{DD} tolerance) mode (POM11 = 1) by using the port output mode register 1 (POM1) (see **4.3 Registers Controlling Port Function** for details). When communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POM10 = 1) also for the clock input/output pin (SCL20) (see **4.4.4 Connecting to external device with different potential (2.5 V)** for details).

Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

(1) Register setting

Figure 12-88. Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC20)

(a) Serial output register 1(SO1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10 0/1	0	0	0	0	1	SO12 ×	1	SO10 0/1

Start condition is generated by manipulating the SO10 bit.

(b) Serial output enable register 1 (SOE1) … Sets only the bits of the target channel.

	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	SOE12 ×	0	SOE10 0/1

SOE10 = 0 until the start condition is generated, and SOE10 = 1 after generation.

(c) Serial channel start register 1 (SS1) … 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
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13 ×	SS12 ×	SS11 ×	SS10 0/1

(d) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10 0/1	CCS10 0	0	0	0	0	0	STS10 0	0	SIS100 0	1	0	0	MD102 1	MD101 0	MD100 0

Operation mode of channel 0
0: Transfer end interrupt

(e) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10 1	RXE10 0	DAP10 0	CKP10 0	0	EOC10 0	PTC101 0	PTC100 0	DIR10 0	0	SLC101 0	SLC100 1	0	DLS102 1	DLS101 1	DLS100 1

Setting of parity bit
00B: No parity

Setting of stop bit
01B: Appending 1 bit (ACK)

(f) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

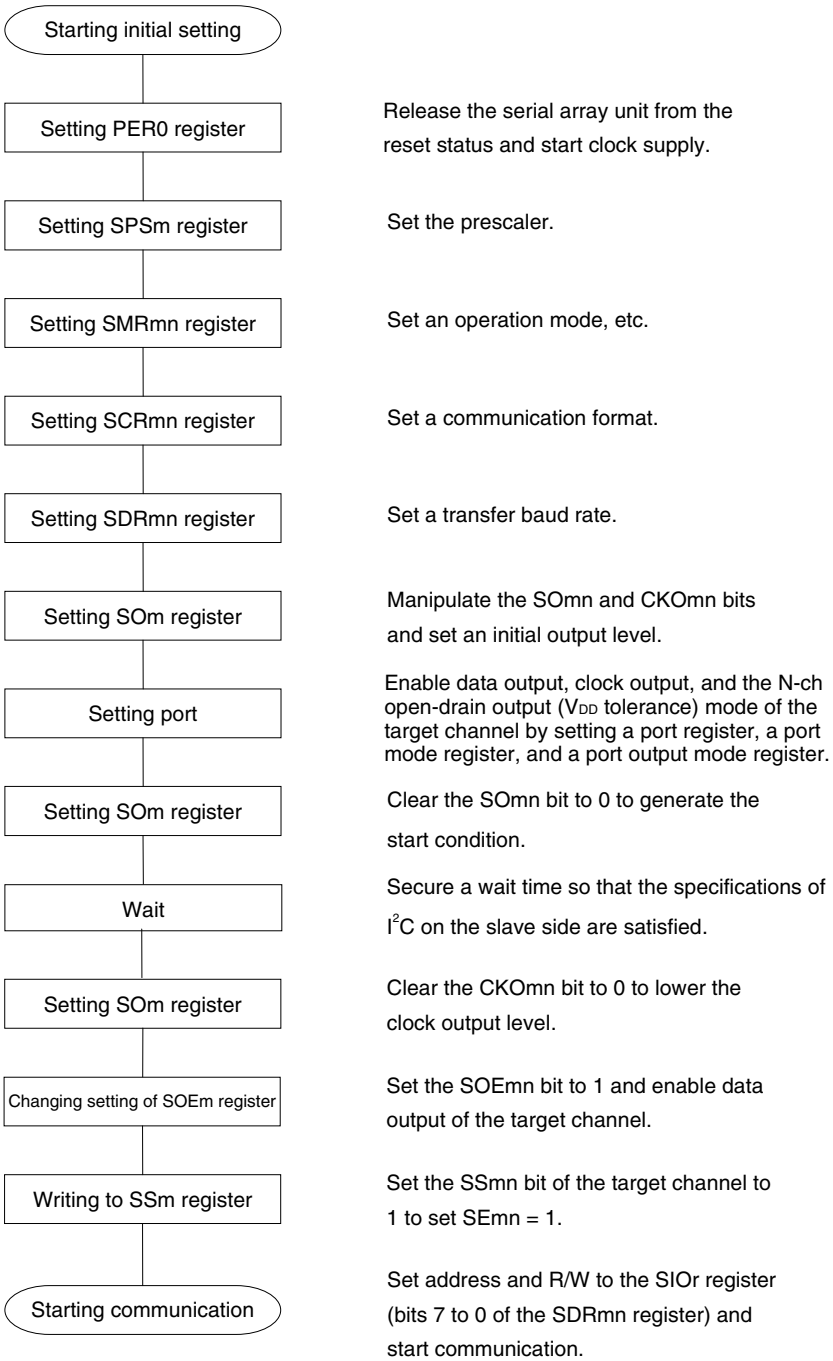
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting							0	Transmit data setting (address + R/W)							

SIO20

Remark □: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-89. Initial Setting Procedure for Address Field Transmission

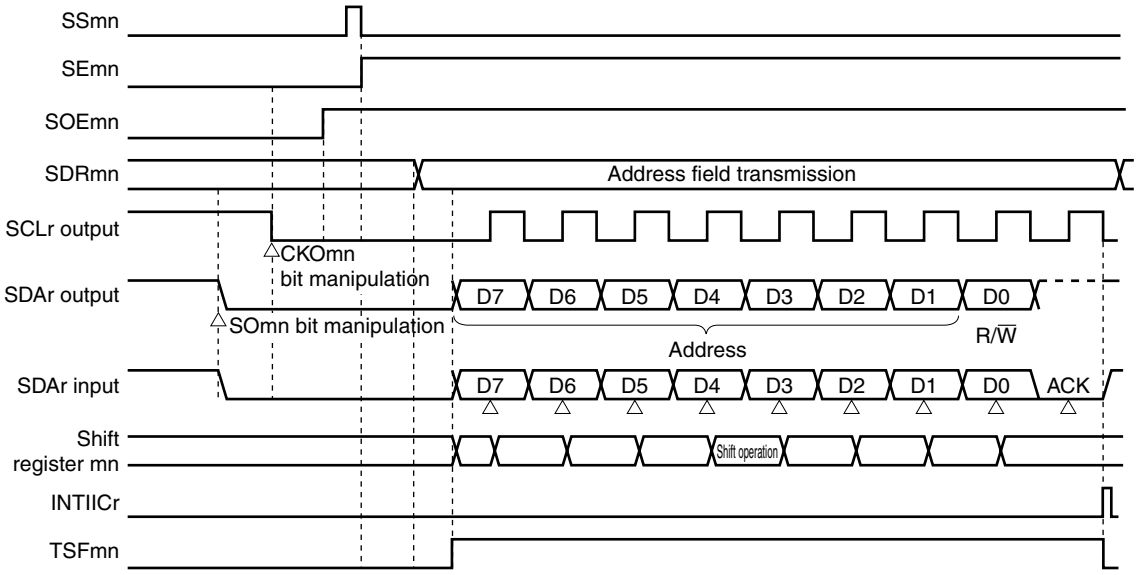


Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

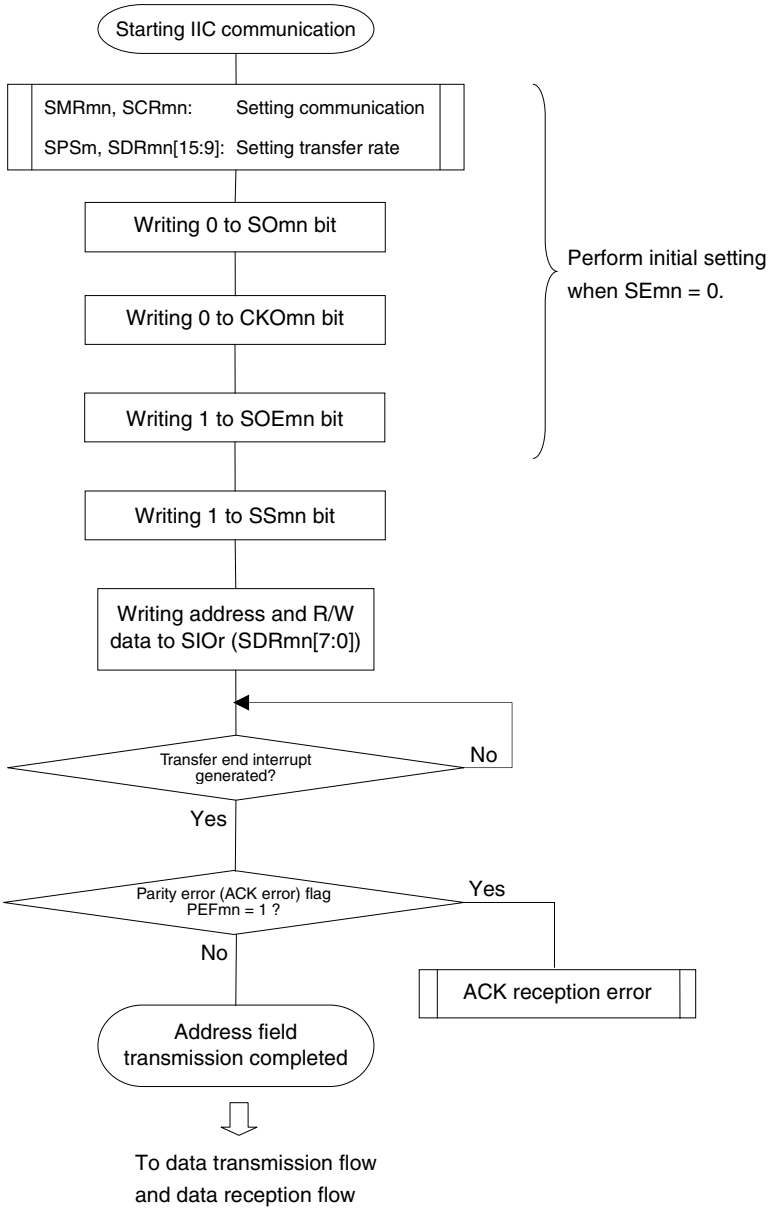
(3) Processing flow

Figure 12-90. Timing Chart of Address Field Transmission



Remark m: Unit number (m = 1), n: Channel number (n = 0), r: IIC number (r = 20), mn = 10

Figure 12-91. Flowchart of Address Field Transmission



Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

12.7.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC20
Target channel	Channel 0 of SAU1
Pins used	SCL20, SDA20[Note]
Interrupt	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Parity error detection flag (PEFmn)
Transfer data length	8 bits
Transfer rate	Max. f_{CLK}/4 [MHz] (SDRmn [15:9] = 1 or more) f_{CLK}: System clock frequency However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)
Data level	Forward output (default: high level)
Parity bit	No parity bit
Stop bit	Appending 1 bit (for ACK reception timing)
Data direction	MSB first

Note To perform communication via simplified I^2C, set the data I/O pin (SDA20) in the N-ch open-drain output (V_{DD} tolerance) mode (POM11 = 1) by using the port output mode register 1 (POM1) (see **4.3 Registers Controlling Port Function** for details). When communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POM10 = 1) also for the clock input/output pin (SCL20) (see **4.4.4 Connecting to external device with different potential (2.5 V)** for details).

Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

(1) Register setting

Figure 12-92. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC20)

(a) Serial output register 1 (SO1) … 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
SO1	0	0	0	0	1	1	1	CKO10 0/1Note	0	0	0	0	1	SO12 ×	1	SO10 0/1Note

(b) Serial output enable register 1 (SOE1) … 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
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12 ×	0	SOE10 0/1

(c) Serial channel start register 1 (SS1) … 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
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13 ×	SS12 ×	SS11 ×	SS10 0/1

(d) Serial mode register 10 (SMR10) … 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
SMR10	CKS10 0/1	CCS10 0	0	0	0	0	0	STS10 0	0	SIS100 0	1	0	0	MD102 1	MD101 0	MD100 0

(e) Serial communication operation setting register 10 (SCR10) … Do not manipulate the bits of this register, except the TXE10 and RXE10 bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10 1	RXE10 0	DAP10 0	CKP10 0	0	EOC10 0	PTC101 0	PTC100 0	DIR10 0	0	SLC101 0	SLC100 1	0	DLS100 1	DLS100 1	DLS100 1

(f) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting							0	Transmit data setting							

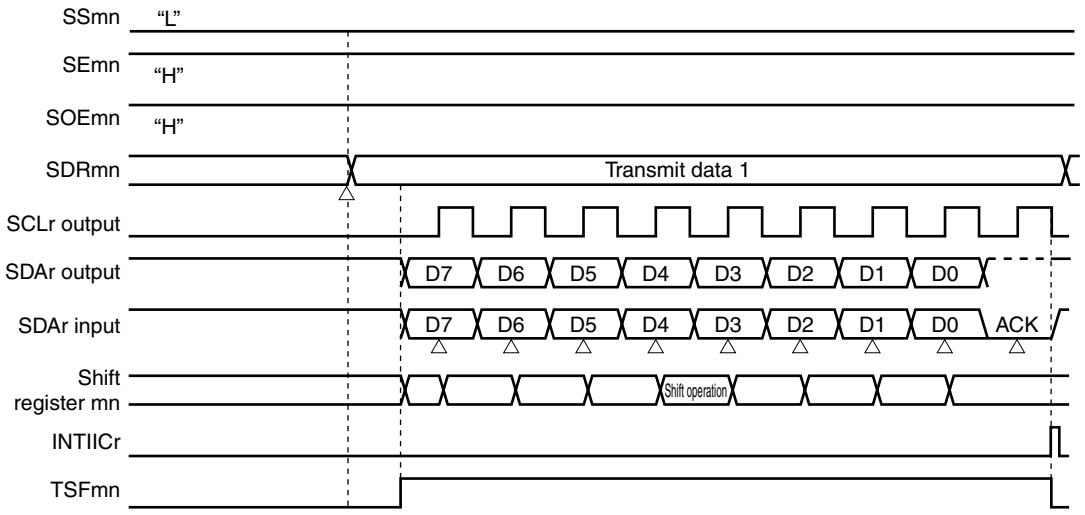
SIO20

Note The value varies depending on the communication data during communication operation.

Remark m: Unit number (m = 1), n: Channel number (n = 0), r: IIC number (r = 20)
□: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

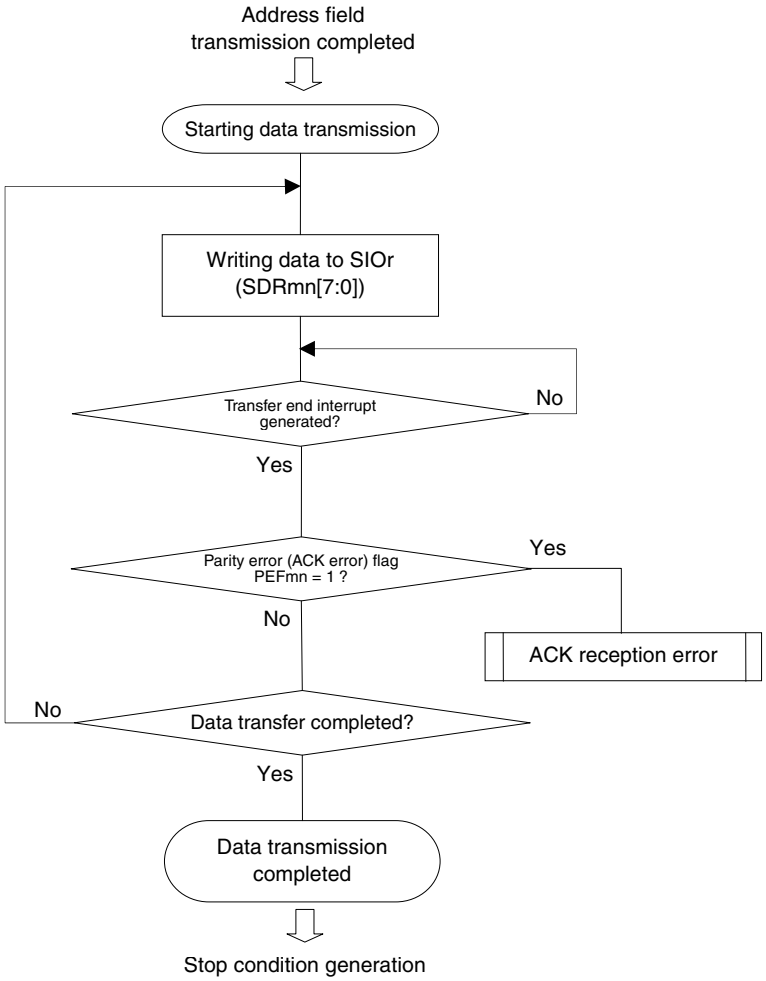
(2) Processing flow

Figure 12-93. Timing Chart of Data Transmission



Remark m: Unit number (m = 1), n: Channel number (n = 0), r: IIC number (r = 20), mn = 10

Figure 12-94. Flowchart of Data Transmission



Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

12.7.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC20
Target channel	Channel 0 of SAU1
Pins used	SCL20, SDA20[Note]
Interrupt	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	None
Transfer data length	8 bits
Transfer rate	Max. f_{CLK}/4 [MHz] (SDRmn [15:9] = 1 or more) f_{CLK}: System clock frequency However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)
Data level	Forward output (default: high level)
Parity bit	No parity bit
Stop bit	Appending 1 bit (ACK transmission)
Data direction	MSB first

Note To perform communication via simplified I^2C, set the data I/O pin (SDA20) in the N-ch open-drain output (V_{DD} tolerance) mode (POM11 = 1) by using the port output mode register 1 (POM1) (see **4.3 Registers Controlling Port Function** for details). When communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POM10 = 1) also for the clock input/output pin (SCL20) (see **4.4.4 Connecting to external device with different potential (2.5 V)** for details).

Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

(1) Register setting

Figure 12-95. Example of Contents of Registers for Data Reception of Simplified I²C (IIC20)

(a) Serial output register 1 (SO1) … 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
SO1	0	0	0	0	1	1	1	CKO10 0/1Note	0	0	0	0	1	SO12 ×	1	SO10 0/1Note

(b) Serial output enable register 1 (SOE1) … 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
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12 ×	0	SOE10 0/1

(c) Serial channel start register 1 (SS1) … 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
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13 ×	SS12 ×	SS11 ×	SS10 0/1

(d) Serial mode register 10 (SMR10) … 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
SMR10	CKS10 0/1	CCS10 0	0	0	0	0	0	STS10 0	0	SIS100 0	1	0	0	MD102 1	MD101 0	MD100 0

(e) Serial communication operation setting register 10 (SCR10) … Do not manipulate the bits of this register, except the TXE10 and RXE10 bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10 0	RXE10 1	DAP10 0	CKP10 0	0	EOC10 0	PTC101 0	PTC100 0	DIR10 0	0	SLC101 0	SLC100 1	0	DLS102 1	DLS101 1	DLS100 1

(f) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting							0	Dummy transmit data setting (FFH)							

SIO20

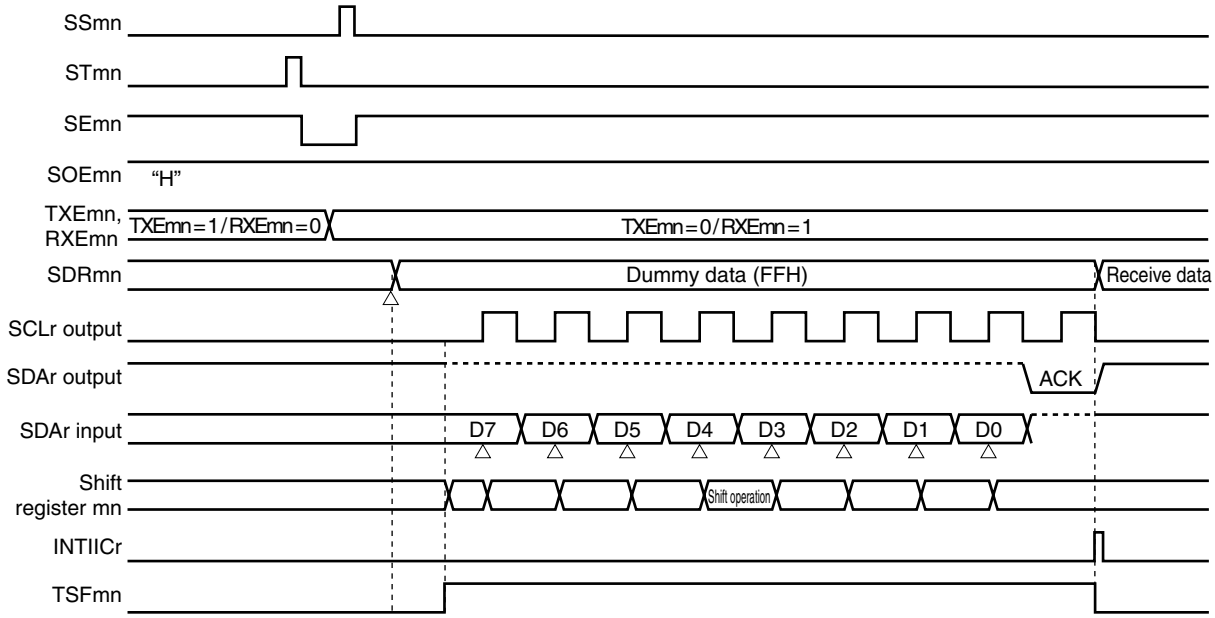
Note The value varies depending on the communication data during communication operation.

Remark □: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

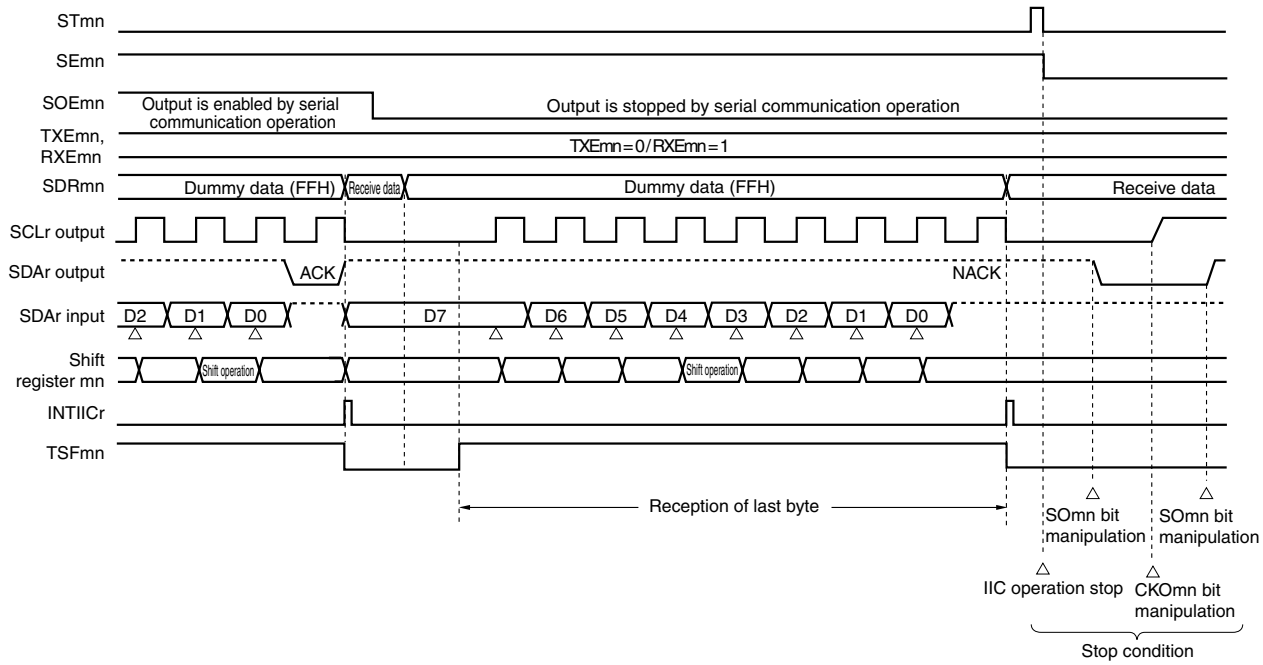
(2) Processing flow

Figure 12-96. Timing Chart of Data Reception

(a) When starting data reception

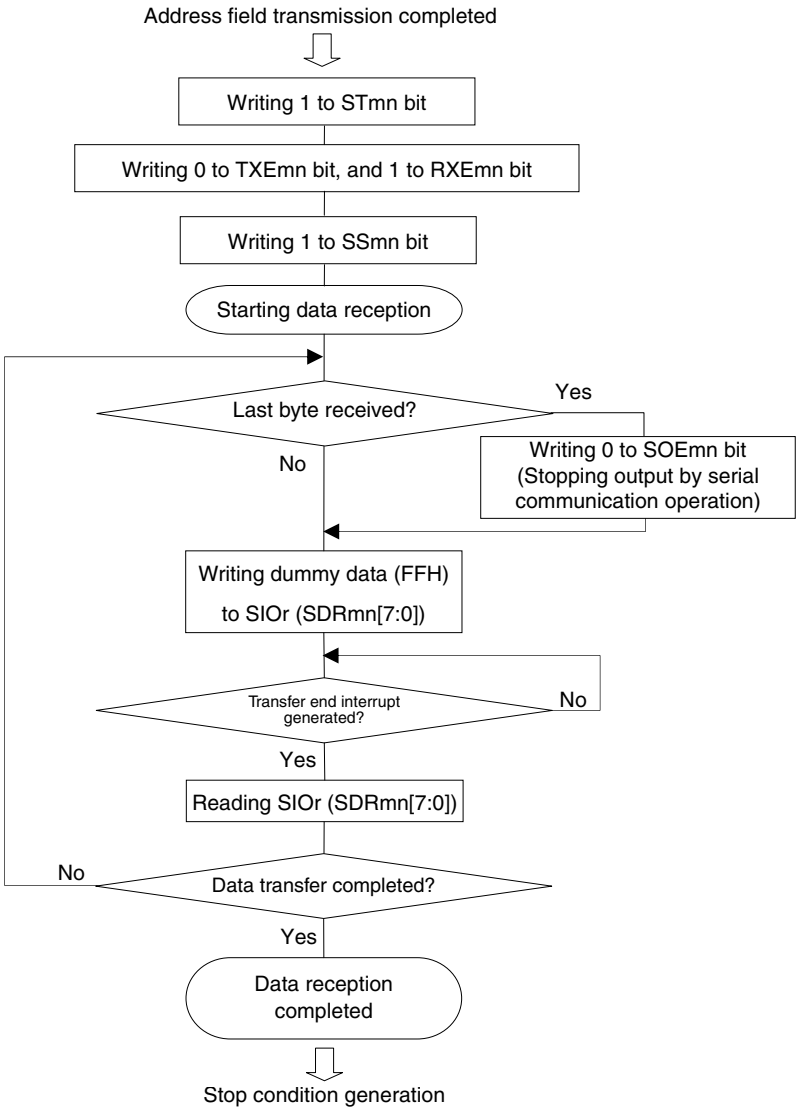


(b) When receiving last data



Remark m: Unit number (m = 1), n: Channel number (n = 0), r: IIC number (r = 20), mn = 10

Figure 12-97. Flowchart of Data Reception



Caution ACK is also output when the last data is received. Communication is then completed by setting "1" to the STmn bit to stop operation and generating a stop condition.

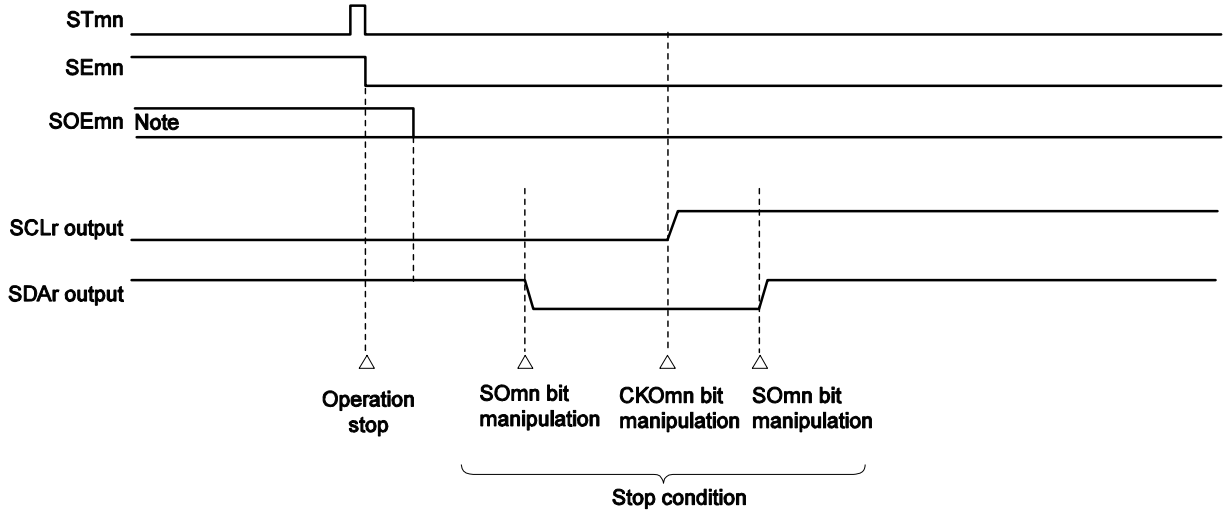
Remark m: Unit number (m = 1), n: Channel number (n = 0), r: IIC number (r = 20), mn = 10

12.7.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

(1) Processing flow

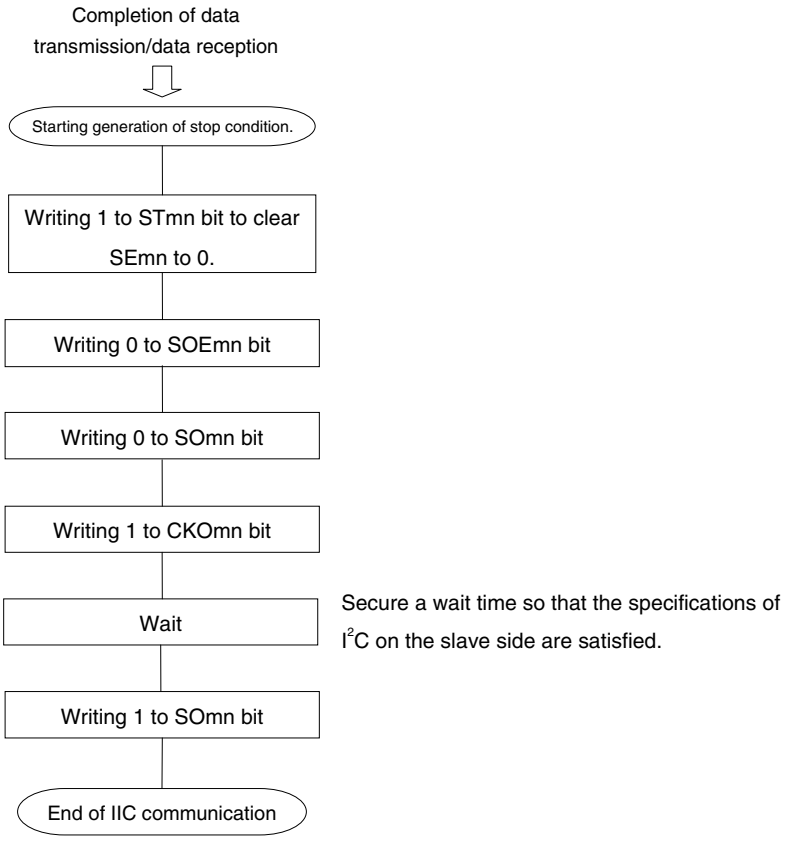
Figure 12-98. Timing Chart of Stop Condition Generation



Note During the receive operation, the SOEmn bit is set to 0 before receiving the last data.

Remark m: Unit number (m = 1), n: Channel number (n = 0), r: IIC number (r = 20), mn = 10

Figure 12-99. Flowchart of Stop Condition Generation



Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

12.7.5 Calculating transfer rate

The transfer rate for simplified I^2C (IIC20) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (MCK) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

Caution Setting SDRmn [15:9] = 0000000B is prohibited. Set SDRmn[15:9] to 0000001B or greater.

Remarks 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000000B to 1111111B) and therefore is 0 to 127.

2. m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 12-4. Selection of operation clock

SMRmn Register	SPSm Register								Operation Clock (MCK)[Note 1]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	313 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM03 if m = 1[Note 2]	
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	313 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM03 if m = 1[Note 2]	
Other than above									Setting prohibited	

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (STm = 000FH) the operation of the serial array unit (SAUm). When selecting INTTM03 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAU can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting TIS03 (if m = 1) of the TIS0 register to 1) and selecting INTTM03 by using the SPSm register in channel 3 of TAU0. When changing f_{CLK}, however, SAUm and TAU0 must be stopped as described in Note 1 above.

Remarks 1. X: don't care

2. m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

Here is an example of setting an IIC transfer rate where MCK = f_{CLK} = 20 MHz.

IIC Transfer Mode (Desired Transfer Rate)	f_{CLK} = 20 MHz			
	Operation Clock (MCK)	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	f_{CLK}	99	100 kHz	0.0%
400 kHz	f_{CLK}	24	400 kHz	0.0%

12.8 Processing Procedure in Case of Error

The processing procedure to be followed if an error of each type occurs is described in Figures 12-100 to 12-102.

Figure 12-100. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads SDRmn register. →	▶ BFF = 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register. →	▶ Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 12-101. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads SDRmn register. →	▶ BFF = 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register. →	▶ 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 STmn bit to 1. →	▶ SEmn = 0, and channel n stops operation.	
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 SSmn bit to 1. →	▶ SEmn = 1, and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 12-102. Processing Procedure in Case of Parity Error (ACK error) in Simplified I^2C Mode

Software Manipulation	Hardware Status	Remark
Reads SDRmn register. →	BFF = 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register. →	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 STmn bit to 1. →	SEmn = 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 SSmn bit to 1. →	SEmn = 1, and channel n is enabled to operate.	

Remark m: Unit number (m = 1), n: Channel number (n = 0), mn = 10

12.9 Relationship Between Register Settings and Pins

Tables 12-5 to 12-10 show the relationship between register settings and pins for each channel of serial array units 0 and 1.

Table 12-5. Relationship between register settings and pins (Channel 0 of unit 0: UART0 transmission)

SE00 Note 1	MD002	MD001	SOE00	SO00	CKO00	TXE00	RXE00	PM80	P80	PM81 Note 2	P81 Note 2	PM82	P82	Operation mode	Pin Function	
															RxD0/INTP9/P81 Note 2	TxD0/P82
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	INTP9/P81	P82
	0	1													RxD0/INTP9/ P81	
1	0	1	1	0/1 Note 4	1	1	0	× Note 3	× Note 3	× Note 3	× Note 3	0	1	UART0 transmission Note 5	RxD0/INTP9/ P81	TxD0

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.

2. When channel 1 of unit 0 is set to UART0 reception, this pin becomes an RxD0 function pin (refer to **Table 12-6**). In this case, operation stop mode or UART0 transmission must be selected for channel 0 of unit 0.

3. This pin can be set as a port function pin.

4. This is 0 or 1, depending on the communication operation. For details, refer to **12.3 (12) Serial output register m (SOm)**.

5. When using UART0 transmission and reception in a pair, set channel 1 of unit 0 to UART0 reception (refer to **Table 12-6**).

Remark ×: don't care

Table 12-6. Relationship between register settings and pins (Channel 1 of unit 0: UART0 reception)

SE01 Note 1	MD012	MD011	SOE01	SO01	CKO01	TXE01	RXE01	PM75	P75	PM76	P76	PM77	P77	PM81 Note 2	P81 Note 2	Operation mode	Pin Function RxD0/INTP9/P81 Note 2
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	INTP9/P80
	0	1															
1	0	1	0	1	1	0	1	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	1	×	UART0 reception Notes 4, 5	RxD0

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.

2. When channel 1 of unit 0 is set to UART0 reception, this pin becomes an RxD0 function pin. In this case, set channel 0 of unit 0 to operation stop mode or UART0 transmission (refer to **Table 12-5**).

3. This pin can be set as a port function pin.

4. When using UART0 transmission and reception in a pair, set channel 0 of unit 0 to UART0 transmission (refer to **Table 12-5**).

5. The SMR00 register of channel 0 of unit 0 must also be set during UART0 reception. For details, refer to **12.5.2 (1) Register setting**.

Remark ×: don't care

Table 12-7. Relationship between register settings and pins
(Channel 0 of unit 1: CSI20, UART2 transmission, IIC20)

SE10 Note 1	MD102	MD101	SOE10	SO10	CKO10	TXE10	RXE10	PM10	P10	PM11 Note 2	P11 Note 2	PM12	P12	Operation mode	Pin Function: SCK20/SCL20/P10	Pin Function: SI20/SDA20/RxD2/INTP6/P11 Note 2	Pin Function: SO20/TxD2/TO02/P12
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	P10	INTP6/P11	TO02/P12
	0	1														RxD2/INTP6/P11	
	1	0														INTP6/P11	
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	Slave CSI20 reception	$\overline{SCK20}$ (input)	SI20	TO02/P12
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	Slave CSI20 transmission	$\overline{SCK20}$ (input)	INTP6/P11	SO20
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	Slave CSI20 transmission/reception	$\overline{SCK20}$ (input)	SI20	SO20
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	Master CSI20 reception	$\overline{SCK20}$ (output)	SI20	TO02/P12
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	Master CSI20 transmission	$\overline{SCK20}$ (output)	INTP6/P11	SO20
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	Master CSI20 transmission/reception	$\overline{SCK20}$ (output)	SI20	SO20
	0	1	1	0/1 Note 4	1	1	0	× Note 3	× Note 3	× Note 3	× Note 3	0	1	UART2 transmission Note 5	P10	RxD2/INTP6/P11	TxD2
0	1	0	0	0/1 Note 6	0/1 Note 6	0	0	0	1	0	1	× Note 3	× Note 3	IIC20 start condition	SCL20	SDA20	TO02/P12
						1	0										
						0	1										
1			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC20 address field transmission	SCL20	SDA20	TO02/P12
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC20 data transmission	SCL20	SDA20	TO02/P12
			1	0/1 Note 4	0/1 Note 4	0	1	0	1	0	1	× Note 3	× Note 3	IIC20 data reception	SCL20	SDA20	TO02/P12
0			0	0/1 Note 7	0/1 Note 7	0	0	0	1	0	1	× Note 3	× Note 3	IIC20 stop condition	SCL20	SDA20	TO02/P12
						1	0										
						0	1										

(**Notes** and **Remark** are listed on the next page.)

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2 When channel 1 of unit 1 is set to UART2 reception, this pin becomes an RxD2 function pin (refer to **Table 12-10**). In this case, operation stop mode or UART2 transmission must be selected for channel 0 of unit 1.

3. This pin can be set as a port function pin.

4. This is 0 or 1, depending on the communication operation. For details, refer to **12.3 (12) Serial output register m (SOm)**.

5. When using UART2 transmission and reception in a pair, set channel 1 of unit 1 to UART2 reception (refer to **Table 12-8**).

6. Set the CKO10 bit to 1 before a start condition is generated. Clear the SO10 bit from 1 to 0 when the start condition is generated.

7. Set the CKO10 bit to 1 before a stop condition is generated. Clear the SO10 bit from 0 to 1 when the stop condition is generated.

Remark ×: don't care

Table 12-8. Relationship between register settings and pins (Channel 1 of unit 1: UART2 reception)

SE11 Note 1	MD112	MD111	TXE11	RXE11	PM11 Note 2	P11 Note 2	Operation mode	Pin Function
								SI20/SDA20/RxD2/ INTP6/P11 Note 2
0	0	1	0	0	× Note 3	× Note 3	Operation stop mode	SI20/SDA20/INTP6/P11
1	0	1	0	1	1	×	UART2 reception Notes 4, 5	RxD2

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. When channel 1 of unit 1 is set to UART2 reception, this pin becomes an RxD2 function pin. In this case, set channel 0 of unit 1 to operation stop mode or UART2 transmission (refer to **Table 12-7**).
When channel 0 of unit 1 is set to CSI20 or IIC20, this pin cannot be used as an RxD2 function pin. In this case, set channel 1 of unit 1 to operation stop mode.

3. This pin can be set as a port function pin.

4. When using UART2 transmission and reception in a pair, set channel 0 of unit 1 to UART2 transmission (refer to **Table 12-7**).

5. The SMR10 register of channel 0 of unit 1 must also be set during UART2 reception. For details, refer to **12.5.2 (1) Register setting**.

Remark ×: don't care

Table 12-9. Relationship between register settings and pins (Channel 2 of unit 1: UART3 transmission)

SE12 Note 1	MD122	MD121	SOE12	SO12	TXE12	RXE12	PM51	P51	Operation mode	Pin Function
										TxD3/SEG38/P51
0	0	1	0	1	0	0	× Note 2	× Note 2	Operation stop mode	SEG38/P51
1	0	1	1	0/1 Note 3	1	0	0	1	UART3 transmission Note 4	TxD3

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. This pin can be set as a port function pin.

3. This is 0 or 1, depending on the communication operation. For details, refer to **12.3 (12) Serial output register m (SOm)**.

4. When using UART3 transmission and reception in a pair, set channel 3 of unit 1 to UART3 reception (refer to **Table 12-10**).

Table 12-10. Relationship between register settings and pins (Channel 3 of unit 1: UART3 reception)

SE13 Note 1	MD132	MD131	TXE13	RXE13	PM50	P50	Operation mode	Pin Function
								RxD3/SEG39/P50
0	0	1	0	0	× Note 2	× Note 2	Operation stop mode	SEG39/P50
1	0	1	0	1	1	×	UART3 reception Notes 3, 4	RxD3

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. This pin can be set as a port function pin.

3. When using UART3 transmission and reception in a pair, set channel 2 of unit 1 to UART3 transmission (refer to **Table 12-19**).

4. The SMR12 register of channel 2 of unit 1 must also be set during UART3 reception. For details, refer to **12.5.2 (1) Register setting**.

Remark ×: don't care

CHAPTER 13 SERIAL INTERFACE IICA

13.1 Functions of Serial Interface IICA

Serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL0) line and a serial data bus (SDA0) line.

This mode complies with the I^2C bus format and the master device can generated “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the I^2C bus.

Since the SCL0 and SDA0 pins are used for open drain outputs, 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 (INTIICA) 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 WUP bit of IICA control register 1 (IICCTL1).

Figure 13-1 shows a block diagram of serial interface IICA.

Figure 13-1. Block Diagram of Serial Interface IICA

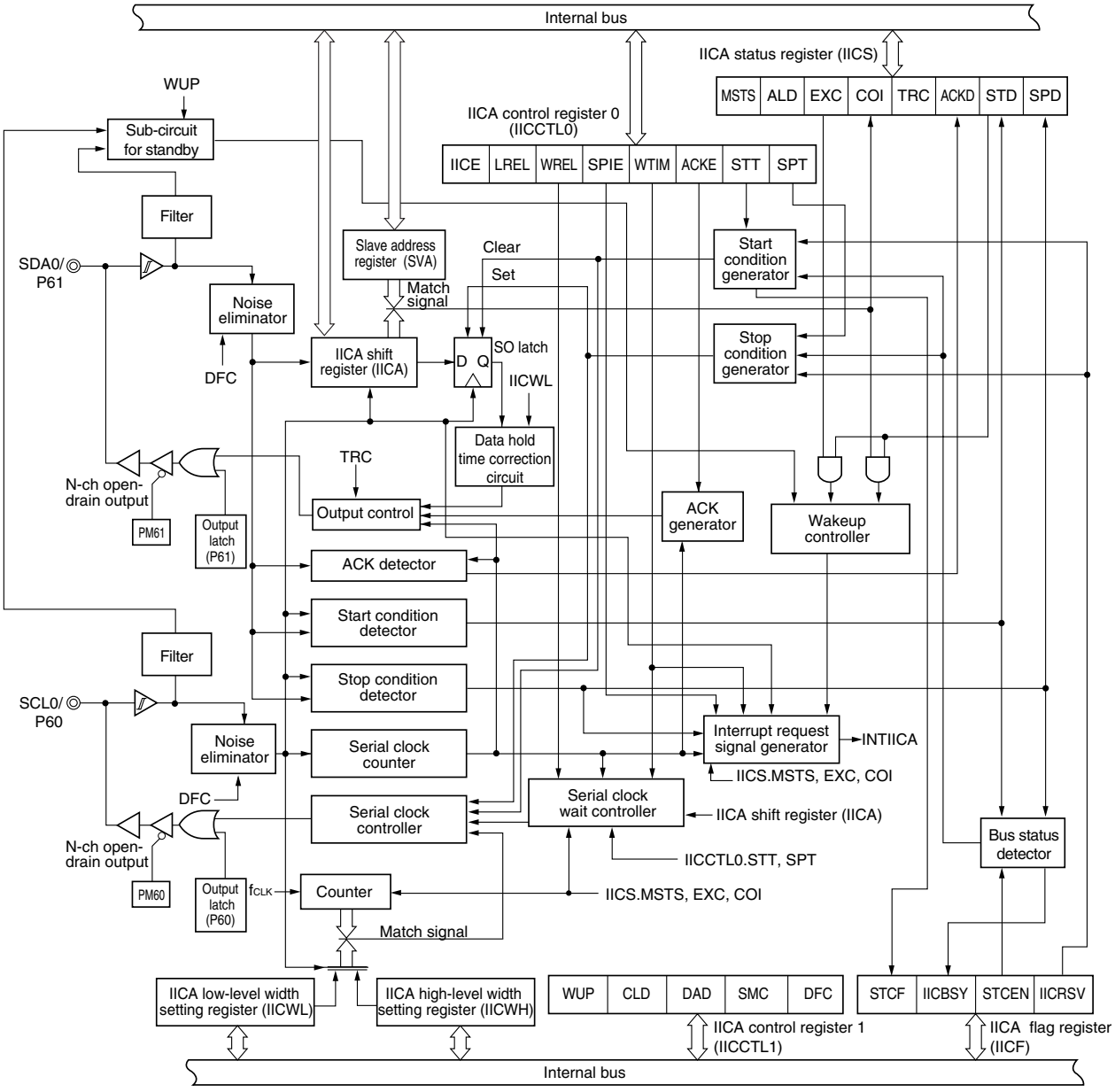
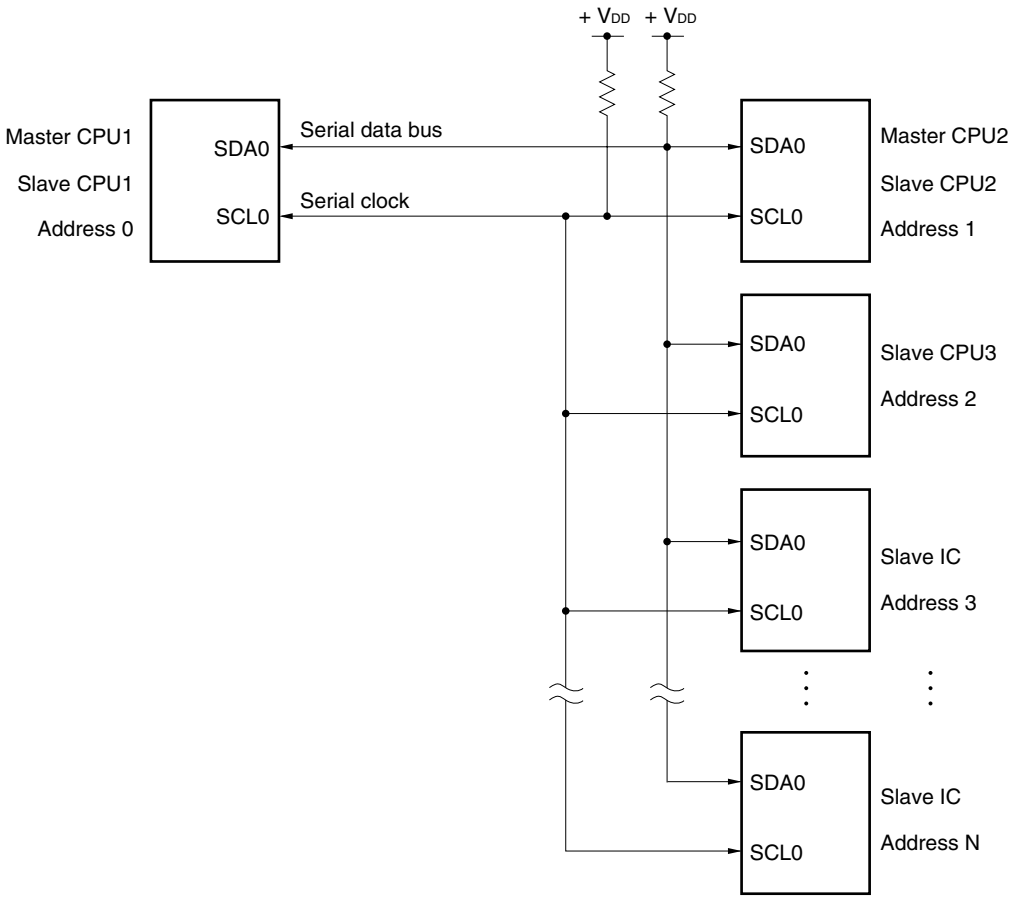


Figure 13-2 shows a serial bus configuration example.

Figure 13-2. Serial Bus Configuration Example Using I²C Bus



13.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 13-1. Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register (IICA) Slave address register (SVA)
Control registers	Peripheral enable register 0 (PER0) IICA control register 0 (IICCTL0) IICA status register (IICS) IICA flag register (IICF) IICA control register 1 (IICCTL1) IICA low-level width setting register (IICWL) IICA high-level width setting register (IICWH) Port mode register 6 (PM6) Port register 6 (P6)

(1) IICA shift register (IICA)

IICA is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock.

IICA can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to IICA.

Cancel the wait state and start data transfer by writing data to IICA during the wait period.

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

Reset signal generation clears IICA to 00H.

Figure 13-3. Format of IICA Shift Register (IICA)

Address: FFF50H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICA								

Cautions 1. Do not write data to IICA during data transfer.

2. Write or read IICA only during the wait period. Accessing IICA in a communication state other than during the wait period is prohibited. When the device serves as the master, however, IICA can be written only once after the communication trigger bit (STT) is set to 1.

3. When communication is reserved, write data to IICA after the interrupt triggered by a stop condition is detected.

(2) Slave address register (SVA)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

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

However, rewriting to this register is prohibited while STD = 1 (while the start condition is detected).

Reset signal generation clears SVA to 00H.

Figure 13-4. Format of Slave Address Register (SVA)

Address: F0234H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA	A6	A5	A4	A3	A2	A1	A0	0Note

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDA0 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICA) when the address received by this register matches the address value set to the slave address register (SVA) 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 (INTIICA).

An I^2C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by WTIM bit)
- Interrupt request generated when a stop condition is detected (set by SPIE bit)

Remark WTIM bit: Bit 3 of IICA control register 0 (IICCTL0)
SPIE bit: Bit 4 of IICA control register 0 (IICCTL0)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCL0 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.

(11) Start condition generator

This circuit generates a start condition when the STT bit is set to 1.

However, in the communication reservation disabled status (IICRSV bit = 1), when the bus is not released (IICBSY bit = 1), start condition requests are ignored and the STCF bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPT bit is set to 1.

(13) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCEN bit.

Remark STT bit: Bit 1 of IICA control register 0 (IICCTL0)
SPT bit: Bit 0 of IICA control register 0 (IICCTL0)
IICRSV bit: Bit 0 of IICA flag register (IICF)
IICBSY bit: Bit 6 of IICA flag register (IICF)
STCF bit: Bit 7 of IICA flag register (IICF)
STCEN bit: Bit 1 of IICA flag register (IICF)

13.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following eight registers.

- Peripheral enable register 0 (PER0)
- IICA control register 0 (IICCTL0)
- IICA flag register (IICF)
- IICA status register (IICS)
- IICA control register 1 (IICCTL1)
- IICA low-level width setting register (IICWL)
- IICA high-level width setting register (IICWH)
- Port mode register 6 (PM6)
- Port register 6 (P6)

(1) Peripheral enable register 0 (PER0)

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

When serial interface IICA is used, be sure to set bit 4 (IICAEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 13-5. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

IICAEN	Control of serial interface IICA input clock
0	Stops supply of input clock. • SFR used by serial interface IICA cannot be written. • Serial interface IICA is in the reset status.
1	Supplies input clock. • SFR used by serial interface IICA can be read/written.

Caution When setting serial interface IICA, be sure to set IICAEN to 1 first. If IICAEN = 0, writing to a control register of serial interface IICA is ignored, and, even if the register is read, only the default value is read.

(2) IICA control register 0 (IICCTL0)

This register is used to enable/stop I^2C operations, set wait timing, and set other I^2C operations.

IICCTL0 can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE, WTIM, and ACKE bits while IICE bit = 0 or during the wait period. These bits can be set at the same time when the IICE bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Figure 13-6. Format of IICA Control Register 0 (IICCTL0) (1/4)

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTL0	IICE	LREL	WREL	SPIE	WTIM	ACKE	STT	SPT

IICE	I^2C operation enable
0	Stop operation. Reset the IICA status register (IICS)[Note 1]. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCL0 and SDA0 lines are at high level.	
Condition for clearing (IICE = 0)	Condition for setting (IICE = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL[Notes 2, 3]	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL0 and SDA0 lines are set to high impedance. The following flags of IICA control register 0 (IICCTL0) and IICA status register (IICS) are cleared to 0. • STT • SPT • MSTS • EXC • COI • TRC • ACKD • STD
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LREL = 0)	Condition for setting (LREL = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL[Notes 2, 3]	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When WREL is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC = 1), the SDA0 line goes into the high impedance state (TRC = 0).	
Condition for clearing (WREL = 0)	Condition for setting (WREL = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICS register, the STCF and IICBSY bits of the IICF register, and the CLD and DAD bits of the IICCTL1 register are reset.

2. The signal of this bit is invalid while IICE is 0.

3. When the LREL and WREL bits are read, 0 is always read.

Caution If the operation of I^2C is enabled (IICE = 1) when the SCL0 line is at high level, the SDA0 line is at low level, and DFC of the IICCTL1 register is 1, a start condition will be inadvertently detected immediately. Immediately after enabling I^2C to operate (IICE = 1), set LREL (1) by using a 1-bit memory manipulation instruction.

Figure 13-6. Format of IICA Control Register 0 (IICCTL0) (2/4)

SPIE[Note 1]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If WUP of the IICCTL1 register is 1, no stop condition interrupt will be generated even if SPIE = 1.	
Condition for clearing (SPIE = 0)	Condition for setting (SPIE = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM[Note 1]	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 ($\overline{ACK}$) 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 (WTIM = 0)	Condition for setting (WTIM = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE[Notes 1, 2]	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDA0 line is set to low level.
Condition for clearing (ACKE = 0)	Condition for setting (ACKE = 1)
• Cleared by instruction • Reset	• Set by instruction

Notes 1. The signal of this bit is invalid while IICE is 0. Set this bit during that period.

2. The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Figure 13-6. Format of IICA Control Register 0 (IICCTL0) (3/4)

STT[Note]	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSY = 0): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation function is enabled (IICRSV = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSV = 1) Even if this bit is set (1), the STT bit is cleared and the STT clear flag (STCF) is set (1). No start condition is generated. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when ACKE has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as SPT.
- Setting STT to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (STT = 0)	Condition for setting (STT = 1)
• Cleared by setting STT to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL = 1 (exit from communications) • When IICE = 0 (operation stop) • Reset	• Set by instruction

Note The signal of this bit is invalid while IICE0 is 0.

Remarks 1. Bit 1 (STT) becomes 0 when it is read after data setting.

2. IICRSV: Bit 0 of IIC flag register (IICF)
STCF: Bit 7 of IIC flag register (IICF)

Figure 13-6. Format of IICA Control Register 0 (IICCTL0) (4/4)

SPT	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer).

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when ACKE has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as STT.
- SPT can be set to 1 only when in master mode.
- When WTIM has been cleared to 0, if SPT 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. WTIM should be changed from 0 to 1 during the wait period following the output of eight clocks, and SPT should be set to 1 during the wait period that follows the output of the ninth clock.
- Setting SPT to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (SPT = 0)	Condition for setting (SPT = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL = 1 (exit from communications) • When IICE = 0 (operation stop) • Reset	• Set by instruction

Caution When bit 3 (TRC) of the IICA status register (IICS) is set to 1, WREL is set to 1 during the ninth clock and wait is canceled, after which TRC is cleared and the SDA0 line is set to high impedance. Release the wait performed while the TRC bit is 1 (transmission status) by writing to the IICA shift register.

Remark Bit 0 (SPT) becomes 0 when it is read after data setting.

(3) IICA status register (IICS)

This register indicates the status of I^2C.

IICS is read by a 1-bit or 8-bit memory manipulation instruction only when STT = 1 and during the wait period.

Reset signal generation clears this register to 00H.

Caution Reading the IICS register while the address match wakeup function is enabled (WUP = 1) in STOP mode is prohibited. When the WUP bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA 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 (SPIE = 1) the interrupt generated by detecting a stop condition and read the IICS register after the interrupt has been detected.

Remark STT: bit 1 of IICA control register 0 (IICCTL0)
WUP: bit 7 of IICA control register 1 (IICCTL1)

Figure 13-7. Format of IICA Status Register (IICS) (1/3)

Address: FFF51H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS	MSTS	ALD	EXC	COI	TRC	ACKD	STD	SPD

MSTS	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS = 0)	Condition for setting (MSTS = 1)
• When a stop condition is detected • When ALD = 1 (arbitration loss) • Cleared by LREL = 1 (exit from communications) • When IICE changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD	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”. MSTS is cleared.

Condition for clearing (ALD = 0)	Condition for setting (ALD = 1)
• Automatically cleared after IICS is read[Note] • When IICE changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than IICS. Therefore, when using the ALD bit, read the data of this bit before the data of the other bits.

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

Figure 13-7. Format of IICA Status Register (IICS) (2/3)

EXC	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC = 0)	Condition for setting (EXC = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 (exit from communications) • When IICE changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

COI	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI = 0)	Condition for setting (COI = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 (exit from communications) • When IICE changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register (SVA)) (set at the rising edge of the eighth clock).

TRC	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDA0 line (valid starting at the falling edge of the first byte’s ninth clock).

Condition for clearing (TRC = 0)	Condition for setting (TRC = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL = 1 (exit from communications) • When the IICE bit changes from 1 to 0 (operation stop) • Cleared by WREL = 1[Note] (wait cancel) • When the ALD bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTS, EXC, COI = 0) <Master> • When “1” is output to the first byte’s LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When “0” is input to the first byte’s LSB (transfer direction specification bit)	<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer) <Slave> • When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)

Note When bit 3 (TRC) of the IICA status register (IICS) is set to 1 (transmission status), bit 5 (WREL) of IICA control register 0 (IICCTL0) is set to 1 during the ninth clock and wait is canceled, after which the TRC bit is cleared (reception status) and the SDA0 line is set to high impedance. Release the wait performed while the TRC bit is 1 (transmission status) by writing to the IICA shift register.

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

Figure 13-7. Format of IICA Status Register (IICS) (3/3)

ACKD	Detection of acknowledge ($\overline{ACK}$)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKD = 0)	Condition for setting (ACKD = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL = 1 (exit from communications) • When IICE changes from 1 to 0 (operation stop) • Reset	• After the SDA0 line is set to low level at the rising edge of SCL0's ninth clock

STD	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 = 0)	Condition for setting (STD = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL = 1 (exit from communications) • When IICE changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD	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 = 0)	Condition for setting (SPD = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When IICE changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

(4) IICA flag register (IICF)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.

IICF can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STCF and IICBSY bits are read-only.

The IICRSV bit can be used to enable/disable the communication reservation function.

STCEN can be used to set the initial value of the IICBSY bit.

IICRSV and STCEN can be written only when the operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) = 0). When operation is enabled, the IICF register can be read.

Reset signal generation clears this register to 00H.

Figure 13-8. Format of IICA Flag Register (IICF)

Address: FFF52H After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICF	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	STT clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear STT flag

Condition for clearing (STCF = 0)	Condition for setting (STCF = 1)
• Cleared by STT = 1 • When IICE = 0 (operation stop) • Reset	• Generating start condition unsuccessful and STT cleared to 0 when communication reservation is disabled (IICRSV = 1).

IICBSY	I^2C bus status flag
0	Bus release status (communication initial status when STCEN = 1)
1	Bus communication status (communication initial status when STCEN = 0)

Condition for clearing (IICBSY = 0)	Condition for setting (IICBSY = 1)
• Detection of stop condition • When IICE = 0 (operation stop) • Reset	• Detection of start condition • Setting of IICE when STCEN = 0

STCEN	Initial start enable trigger
0	After operation is enabled (IICE = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN = 0)	Condition for setting (STCEN = 1)
• Cleared by instruction • Detection of start condition • Reset	• Set by instruction

IICRSV	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV = 0)	Condition for setting (IICRSV = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to STCEN only when the operation is stopped (IICE = 0).

2. As the bus release status (IICBSY = 0) is recognized regardless of the actual bus status when STCEN = 1, when generating the first start condition (STT = 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 IICRSV only when the operation is stopped (IICE = 0).

Remark STT: Bit 1 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

(5) IICA control register 1 (IICCTL1)

This register is used to set the operation mode of I^2C and detect the statuses of the SCL0 and SDA0 pins.

IICCTL1 can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD and DAD bits are read-only.

Set the IICCTL1 register, except the WUP bit, while operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) is 0).

Reset signal generation clears this register to 00H.

Figure 13-9. Format of IICA Control Register 1 (IICCTL1) (1/2)

Address: F0231H After reset: 00H R/W[Note 1]

<R>

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	0
IICCTL1	WUP	0	CLD	DAD	SMC	DFC	0	0

WUP	Control of address match wakeup
0	Stops operation of address match wakeup function in STOP mode.
1	Enables operation of address match wakeup function in STOP mode.

To shift to STOP mode when WUP = 1, execute the STOP instruction at least three clocks after setting (1) the WUP bit (see **Figure 13-22 Flow When Setting WUP = 1**).

Clear (0) the WUP bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUP bit. (The wait must be released and transmit data must be written after the WUP bit has been cleared (0).)

The interrupt timing when the address has matched or when an extension code has been received, while WUP = 1, is identical to the interrupt timing when WUP = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUP = 1, a stop condition interrupt is not generated even if the SPIE bit is set to 1.

When WUP = 0 is set by a source other than an interrupt from serial interface IICA, operation as the master device cannot be performed until the subsequent start condition or stop condition is detected. Do not output a start condition by setting (1) the STT bit, without waiting for the detection of the subsequent start condition or stop condition.

Condition for clearing (WUP = 0)	Condition for setting (WUP = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when the MSTS, EXC, and COI bits are “0”, and the STD bit also “0” (communication not entered))[Note 2]

Notes 1. Bits 4 and 5 are read-only.

2. The status of the IICA status register (IICS) must be checked and the WUP bit must be set during the period shown below.

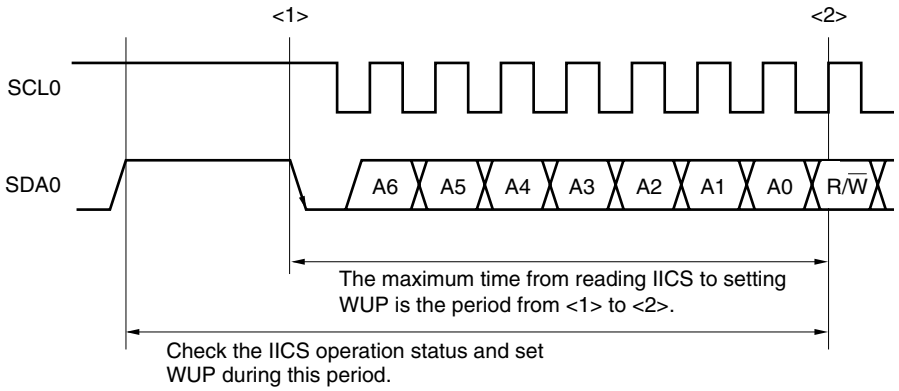


Figure 13-9. Format of IICA Control Register 1 (IICCTL1) (2/2)

CLD	Detection of SCL0 pin level (valid only when IICE = 1)
0	The SCL0 pin was detected at low level.
1	The SCL0 pin was detected at high level.

Condition for clearing (CLD = 0)	Condition for setting (CLD = 1)
• When the SCL0 pin is at low level • When IICE = 0 (operation stop) • Reset	• When the SCL0 pin is at high level

DAD	Detection of SDA0 pin level (valid only when IICE = 1)
0	The SDA0 pin was detected at low level.
1	The SDA0 pin was detected at high level.

Condition for clearing (DAD = 0)	Condition for setting (DAD = 1)
• When the SDA0 pin is at low level • When IICE = 0 (operation stop) • Reset	• When the SDA0 pin is at high level

SMC	Operation mode switching
0	Operates in standard mode.
1	Operates in fast mode.

DFC	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in fast mode. In fast mode, the transfer clock does not vary, regardless of the DFC bit being set (1) or cleared (0). The digital filter is used for noise elimination in fast mode.	

Remark IICE: Bit 7 of IICA control register 0 (IICCTL0)

(6) IICA low-level width setting register (IICWL)

This register is used to set the low-level width of the SCL0 pin signal that is output by serial interface IICA.

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

Set the IICWL register while operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) is 0).

Reset signal generation sets this register to FFH.

Figure 13-10. Format of IICA Low-Level Width Setting Register (IICWL)

Address: F0232H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWL								

(7) IICA high-level width setting register (IICWH)

This register is used to set the high-level width of the SCL0 pin signal that is output by serial interface IICA.

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

Set the IICWH register while operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) is 0).

Reset signal generation sets this register to FFH.

Figure 13-11. Format of IICA High-Level Width Setting Register (IICWH)

Address: F0233H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWH								

Remark For how to set the transfer clock by using the IICWL and IICWH registers, see **13.4.2 Setting transfer clock by using IICWL and IICWH registers**.

(8) Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCL0 pin as clock I/O and the P61/SDA0 pin as serial data I/O, clear PM60 and PM61, and the output latches of P60 and P61 to 0.

Set IICE (bit 7 of IICA control register 0 (IICCTL0)) to 1 before setting the output mode because the P60/SCL0 and P61/SDA0 pins output a low level (fixed) when IICE is 0.

PM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 13-12. Format of Port Mode Register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

13.4 I^2C Bus Mode Functions

13.4.1 Pin configuration

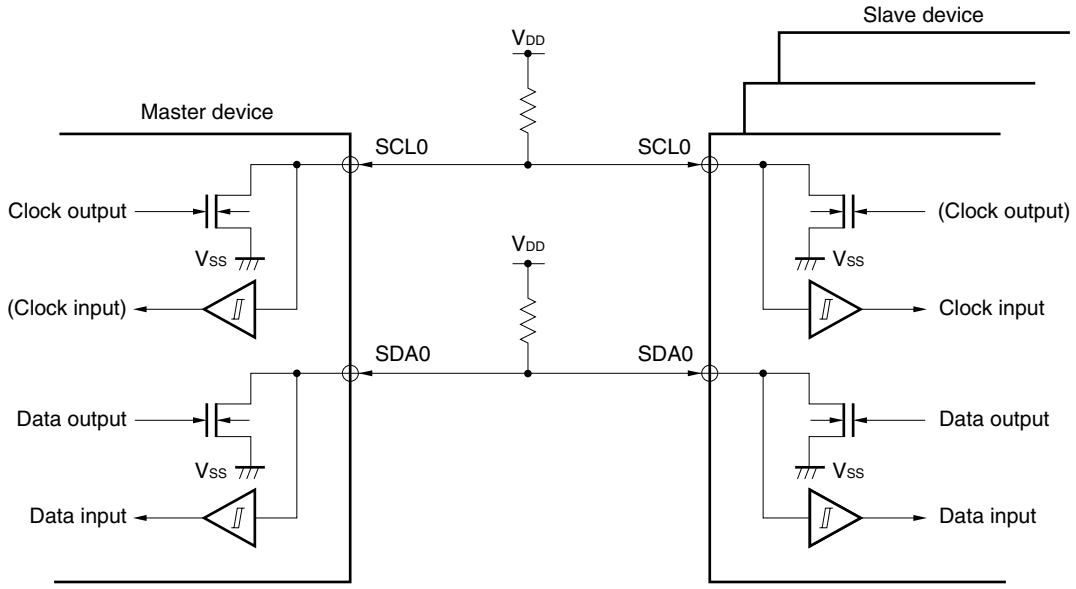
The serial clock pin (SCL0) and serial data bus pin (SDA0) are configured as follows.

(1) SCL0..... 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) SDA0 ..... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

Figure 13-13. Pin Configuration Diagram



13.4.2 Setting transfer clock by using IICWL and IICWH registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{CLK}}{IICWL + IICWH + f_{CLK}(t_R + t_F)}$$

At this time, the optimal setting values of IICWL and IICWH are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICWL = \frac{0.52}{\text{Transfer clock}} \times f_{CLK}$$
$$IICWH = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{CLK}$$

- When the standard mode

$$IICWL = \frac{0.47}{\text{Transfer clock}} \times f_{CLK}$$
$$IICWH = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{CLK}$$

(2) Setting IICWL and IICWH on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$IICWL = 1.3\ \mu s \times f_{CLK}$$
$$IICWH = (1.2\ \mu s - t_R - t_F) \times f_{CLK}$$

- When the standard mode

$$IICWL = 4.7\ \mu s \times f_{CLK}$$
$$IICWH = (5.3\ \mu s - t_R - t_F) \times f_{CLK}$$

Caution Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.

Fast mode: f_{CLK} = 3.5 MHz (MIN.)

Standard mode: f_{CLK} = 1 MHz (MIN.)

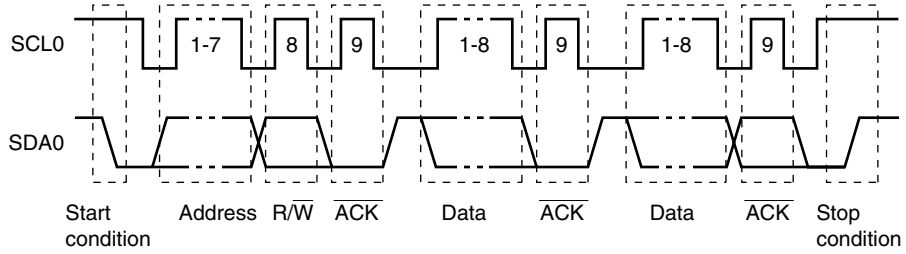
Remarks 1. Calculate the rise time (t_R) and fall time (t_F) of the SDA0 and SCL0 signals separately, because they differ depending on the pull-up resistance and wire load.

2. IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
t_R: SDA0 and SCL0 signal rising times
f_{CLK}: CPU/peripheral hardware clock frequency

13.5 I^2C Bus Definitions and Control Methods

The following section describes the I^2C bus's serial data communication format and the signals used by the I^2C bus. Figure 13-14 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I^2C bus's serial data bus.

Figure 13-14. I^2C Bus Serial Data Transfer Timing



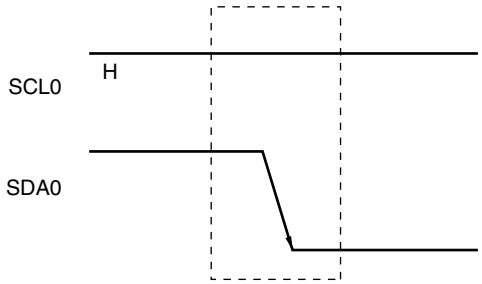
The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCL0) is continuously output by the master device. However, in the slave device, the SCL0's low level period can be extended and a wait can be inserted.

13.5.1 Start conditions

A start condition is met when the SCL0 pin is at high level and the SDA0 pin changes from high level to low level. The start conditions for the SCL0 pin and SDA0 pin are signals that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

Figure 13-15. Start Conditions



A start condition is output when bit 1 (STT) of IICA control register 0 (IICCTL0) is set (1) after a stop condition has been detected (SPD: Bit 0 of the IICA status register (IICS) = 1). When a start condition is detected, bit 1 (STD) of IICS is set (1).

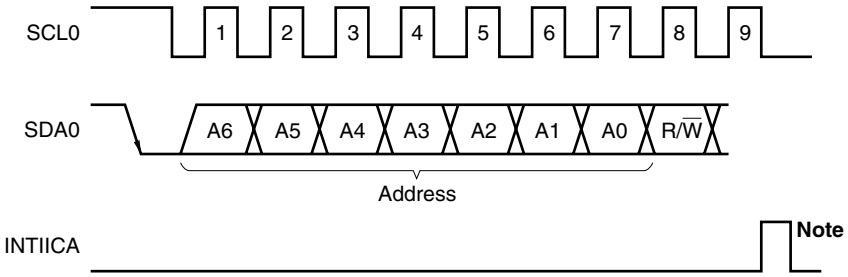
13.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register (SVA). If the address data matches the SVA values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 13-16. Address



Note INTIICA is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **13.5.3 Transfer direction specification** are written to the IICA shift register (IICA). The received addresses are written to IICA.

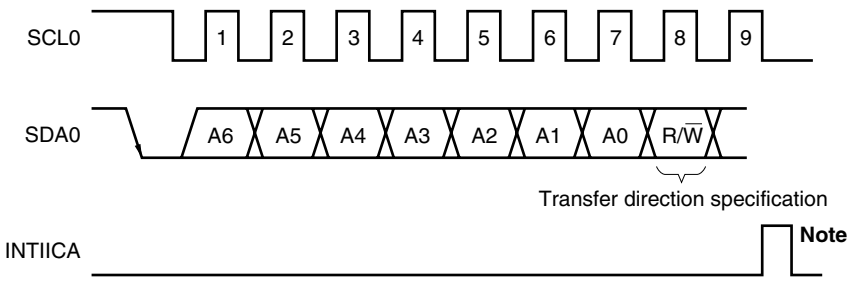
The slave address is assigned to the higher 7 bits of IICA.

13.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit has a value of “0”, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of “1”, it indicates that the master device is receiving data from a slave device.

Figure 13-17. Transfer Direction Specification



Note INTIICA is not issued if data other than a local address or extension code is received during slave device operation.

13.5.4 Acknowledge ($\overline{ACK}$)

$\overline{ACK}$ is used to check the status of serial data at the transmission and reception sides.

The reception side returns $\overline{ACK}$ each time it has received 8-bit data.

The transmission side usually receives $\overline{ACK}$ after transmitting 8-bit data. When $\overline{ACK}$ is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether $\overline{ACK}$ has been detected can be checked by using bit 2 (ACKD) of the IICA status register (IICS).

When the master receives the last data item, it does not return $\overline{ACK}$ and instead generates a stop condition. If a slave does not return $\overline{ACK}$ after receiving data, the master outputs a stop condition or restart condition and stops transmission. If $\overline{ACK}$ is not returned, the possible causes are as follows.

<1> Reception was not performed normally.
<2> The final data item was received.
<3> The reception side specified by the address does not exist.

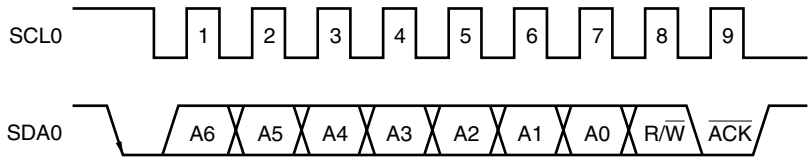
To generate $\overline{ACK}$, the reception side makes the SDA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{ACK}$ is enabled by setting bit 2 (ACKE) of IICA control register 0 (IICCTL0) to 1. Bit 3 (TRC) of the IICS register is set by the data of the eighth bit that follows 7-bit address information. Usually, set ACKE to 1 for reception (TRC = 0).

If a slave can receive no more data during reception (TRC = 0) or does not require the next data item, then the slave must inform the master, by clearing ACKE to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC = 0), it must clear ACKE to 0 so that $\overline{ACK}$ is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 13-18. $\overline{ACK}$



When the local address is received, $\overline{ACK}$ is automatically generated, regardless of the value of ACKE. When an address other than that of the local address is received, $\overline{ACK}$ is not generated (NACK).

When an extension code is received, $\overline{ACK}$ is generated if ACKE is set to 1 in advance.

How $\overline{ACK}$ is generated when data is received differs as follows depending on the setting of the wait timing.

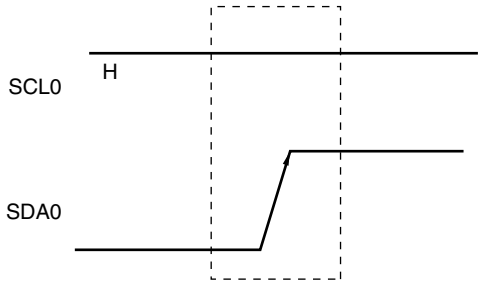
- When 8-clock wait state is selected (bit 3 (WTIM) of IICCTL0 register = 0):
  By setting ACKE to 1 before releasing the wait state, $\overline{ACK}$ is generated at the falling edge of the eighth clock of the SCL0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM) of IICCTL0 register = 1):
  $\overline{ACK}$ is generated by setting ACKE to 1 in advance.

13.5.5 Stop condition

When the SCL0 pin is at high level, changing the SDA0 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 13-19. Stop Condition



A stop condition is generated when bit 0 (SPT) of IICA control register 0 (IICCTL0) is set to 1. When the stop condition is detected, bit 0 (SPD) of the IICA status register (IICS) is set to 1 and INTIICA is generated when bit 4 (SPIE) of IICCTL0 is set to 1.

13.5.6 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 SCL0 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 13-20. 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 ACKE = 1)

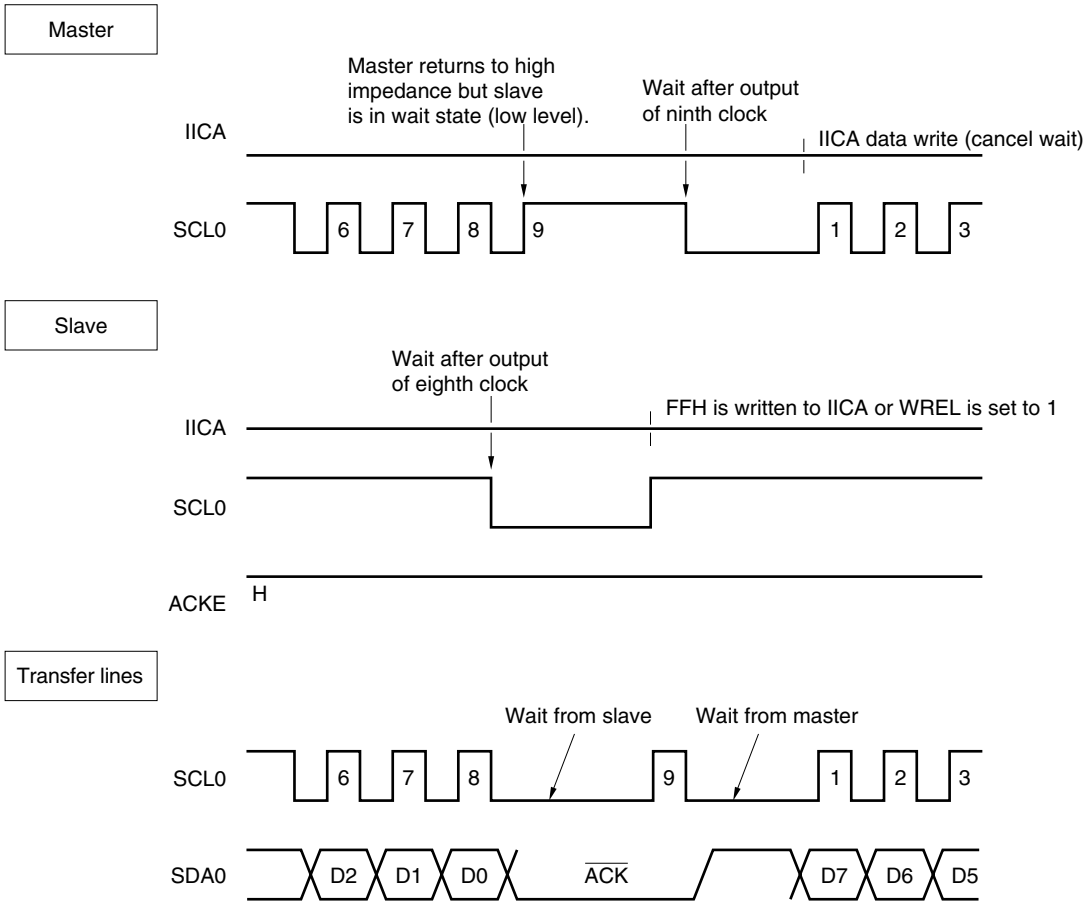
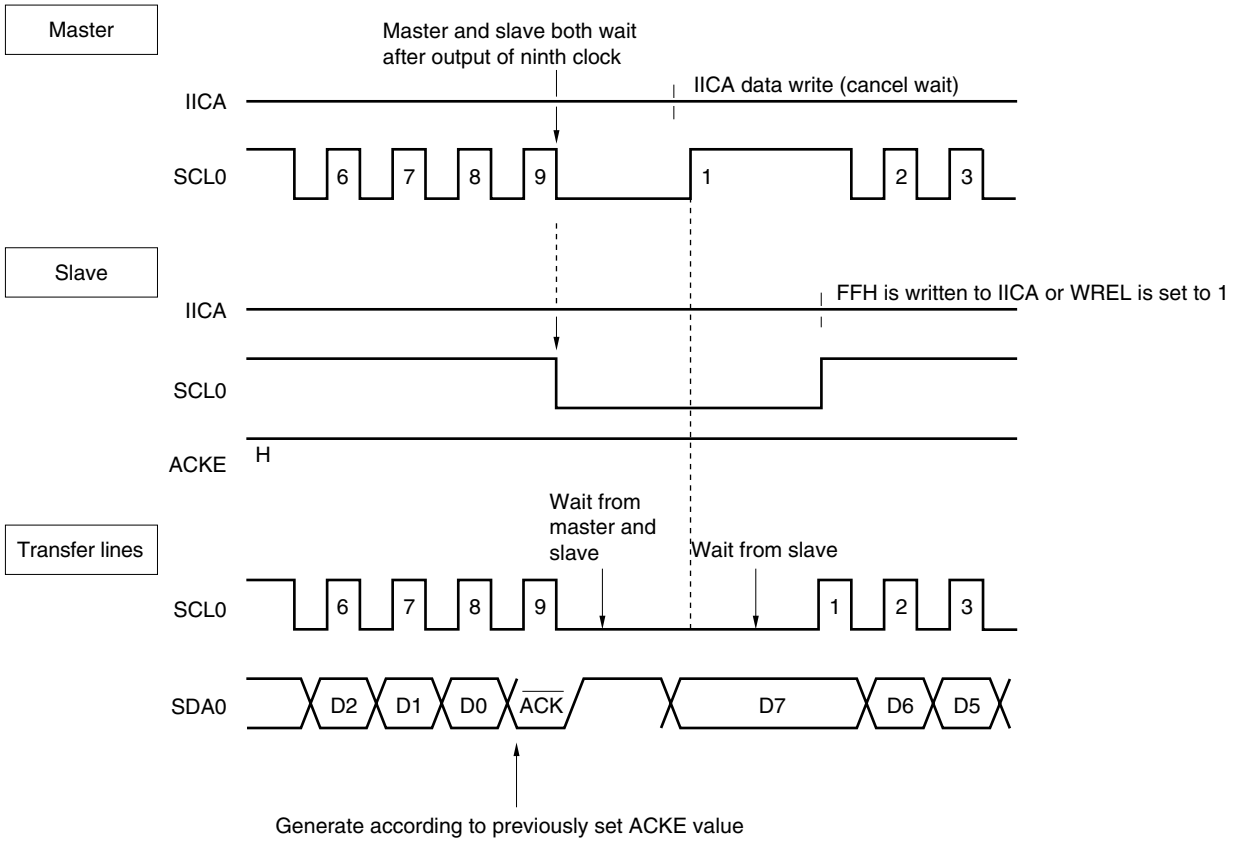


Figure 13-20. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE = 1)



Remark ACKE: Bit 2 of IICA control register 0 (IICCTL0)
WREL: Bit 5 of IICA control register 0 (IICCTL0)

A wait may be automatically generated depending on the setting of bit 3 (WTIM) of IICA control register 0 (IICCTL0).

Normally, the receiving side cancels the wait state when bit 5 (WREL) of the IICCTL0 register is set to 1 or when FFH is written to the IICA shift register (IICA), and the transmitting side cancels the wait state when data is written to the IICA register.

- By setting bit 1 (STT) of IICCTL0 to 1
- By setting bit 0 (SPT) of IICCTL0 to 1

13.5.7 Canceling wait

The I^2C usually cancels a wait state by the following processing.

- Writing data to IICA shift register (IICA)
- Setting bit 5 (WREL) of IICA control register 0 (IICCTL0) (canceling wait)
- Setting bit 1 (STT) of IICCTL0 register (generating start condition)[Note]
- Setting bit 0 (SPT) of IICCTL0 register (generating stop condition)[Note]

Note Master only

When the above wait canceling processing is executed, the I^2C cancels the wait state and communication is resumed.

To cancel a wait state and transmit data (including addresses), write the data to IICA.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL) of IICA control register 0 (IICCTL0) to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT) of IICCTL0 to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT) of IICCTL0 to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to IICA after canceling a wait state by setting WREL to 1, an incorrect value may be output to SDA0 because the timing for changing the SDA0 line conflicts with the timing for writing IICA.

In addition to the above, communication is stopped if IICE is cleared to 0 when communication has been aborted, so that the wait state can be canceled.

If the I^2C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LREL) of IICCTL0, so that the wait state can be canceled.

Caution If a processing to cancel a wait state executed when WUP (bit 7 of IICA control register 1 (IICCTL1)) = 1, the wait state will not be canceled.

13.5.8 Interrupt request (INTIICA) generation timing and wait control

The setting of bit 3 (WTIM) of IICA control register 0 (IICCTL0) determines the timing by which INTIICA is generated and the corresponding wait control, as shown in Table 13-2.

Table 13-2. INTIICA Generation Timing and Wait Control

WTIM	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9 Notes 1, 2	8 Note 2	8 Note 2	9	8	8
1	9 Notes 1, 2	9 Note 2	9 Note 2	9	9	9

Notes 1. The slave device's INTIICA 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 (SVA).
At this point, $\overline{ACK}$ is generated regardless of the value set to IICCTL0's bit 2 (ACKE). For a slave device that has received an extension code, INTIICA occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICA 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 (SVA) and extension code is not received, neither INTIICA 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 WTIM bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to IICA shift register (IICA)
- Setting bit 5 (WREL) of IICA control register 0 (IICCTL0) (canceling wait)
- Setting bit 1 (STT) of IICCTL0 register (generating start condition)Note
- Setting bit 0 (SPT) of IICCTL0 register (generating stop condition)Note

Note Master only.

When an 8-clock wait has been selected (WTIM = 0), the presence/absence of $\overline{ACK}$ generation must be determined prior to wait cancellation.

(5) Stop condition detection

INTIICA is generated when a stop condition is detected (only when SPIE = 1).

13.5.9 Address match detection method

In I^2C 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 (INTIICA) occurs when the address set to the slave address register (SVA) matches the slave address sent by the master device, or when an extension code has been received.

13.5.10 Error detection

In I^2C bus mode, the status of the serial data bus (SDA0) during data transmission is captured by the IICA shift register (IICA) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

13.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC) is set to 1 for extension code reception and an interrupt request (INTIICA) is issued at the falling edge of the eighth clock. The local address stored in the slave address register (SVA) 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 SVA register is set to 11110xx0. Note that INTIICA occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC = 1
- Seven bits of data match: COI = 1

Remark EXC: Bit 5 of IICA status register (IICS)
COI: Bit 4 of IICA status register (IICS)

(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 (LREL) of the IICA control register 0 (IICCTL0) to 1 to set the standby mode for the next communication operation.

Table 13-3. Bit Definitions of Main Extension Code

Slave Address	R/$\overline{W}$ 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 (for address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (for read command issuance after address match)

Remark For extension codes other than the above, refer to THE I^2C-BUS SPECIFICATION published by NXP.

13.5.12 Arbitration

When several master devices simultaneously generate a start condition (when STT is set to 1 before STD 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 (ALD) in the IICA status register (IICS) is set (1) via the timing by which the arbitration loss occurred, and the SCL0 and SDA0 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 ALD = 1 setting that has been made by software.

For details of interrupt request timing, see **13.5.8 Interrupt request (INTIICA) generation timing and wait control**.

Remark STD: Bit 1 of IICA status register (IICS)
STT: Bit 1 of IICA control register 0 (IICCTL0)

Figure 13-21. Arbitration Timing Example

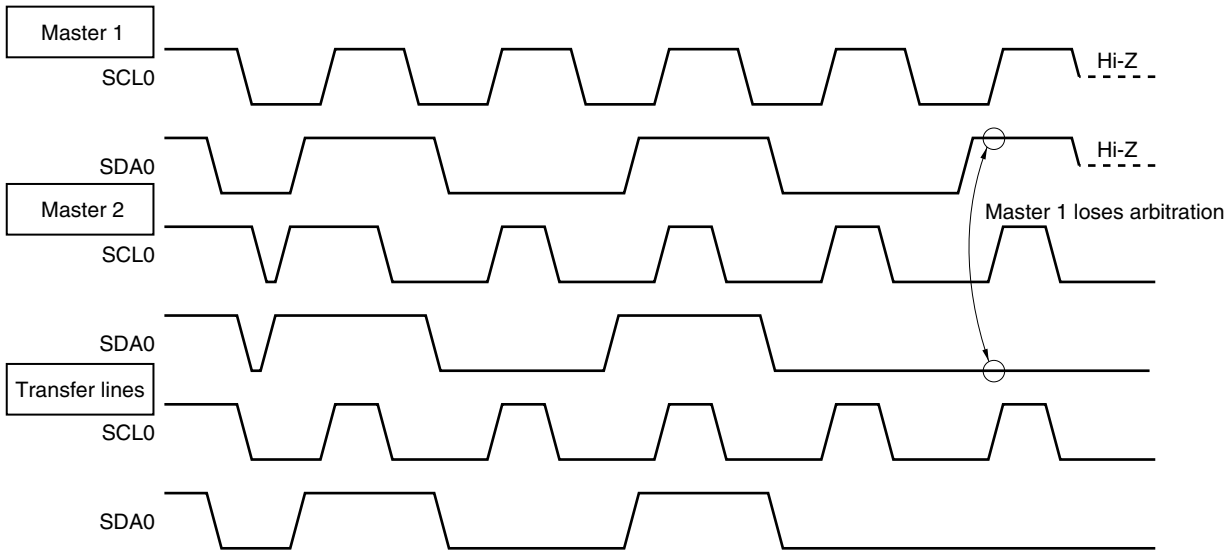


Table 13-4. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when SPIE = 1)[Note 2]
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIE = 1)[Note 2]
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When SCL0 is at low level while attempting to generate a restart condition	

Notes 1. When WTIM (bit 3 of IICA control register 0 (IICCTL0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM = 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 SPIE = 1 for master device operation.

Remark SPIE: Bit 4 of IICA control register 0 (IICCTL0)

13.5.13 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIICA) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIICA 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.

However, when a stop condition is detected, bit 4 (SPIE) of IICA control register 0 (IICCTL0) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

To use the wakeup function in the STOP mode, set WUP to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICA) 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 WUP bit after this interrupt has been generated.

Figure 13-22 shows the flow for setting WUP = 1 and Figure 13-23 shows the flow for setting WUP = 0 upon an address match.

Figure 13-22. Flow When Setting WUP = 1

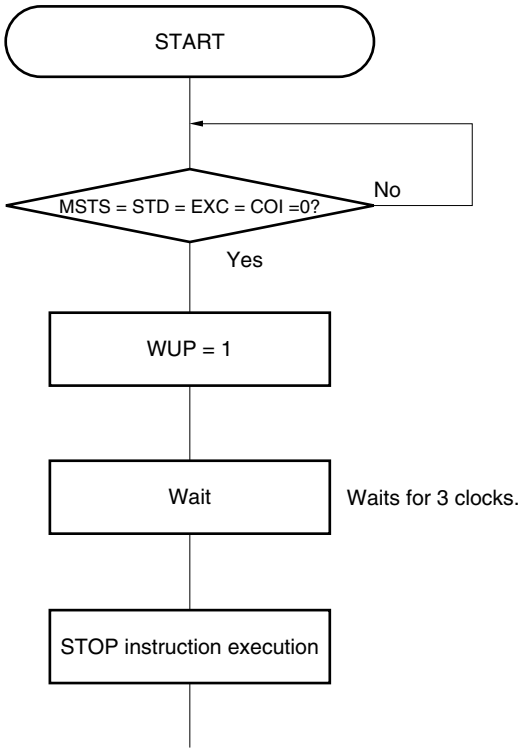
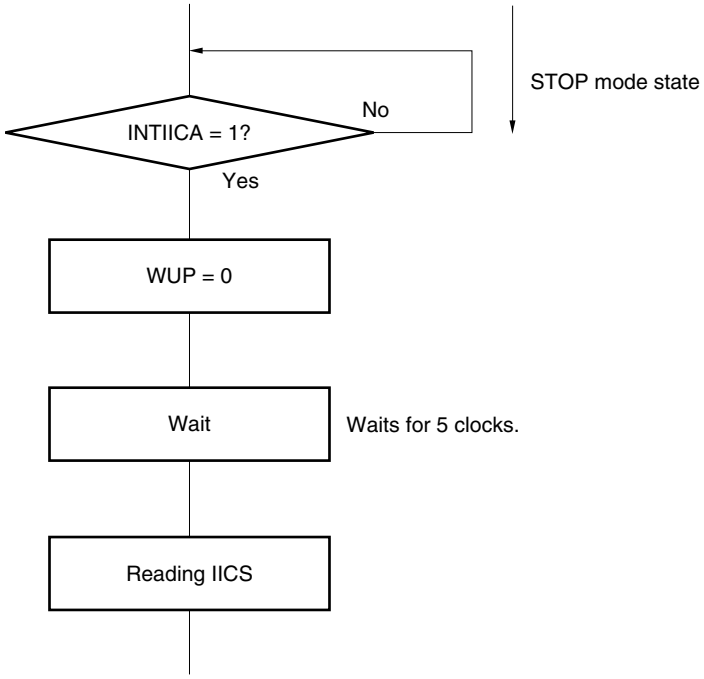


Figure 13-23. Flow When Setting WUP = 0 upon Address Match (Including Extension Code Reception)

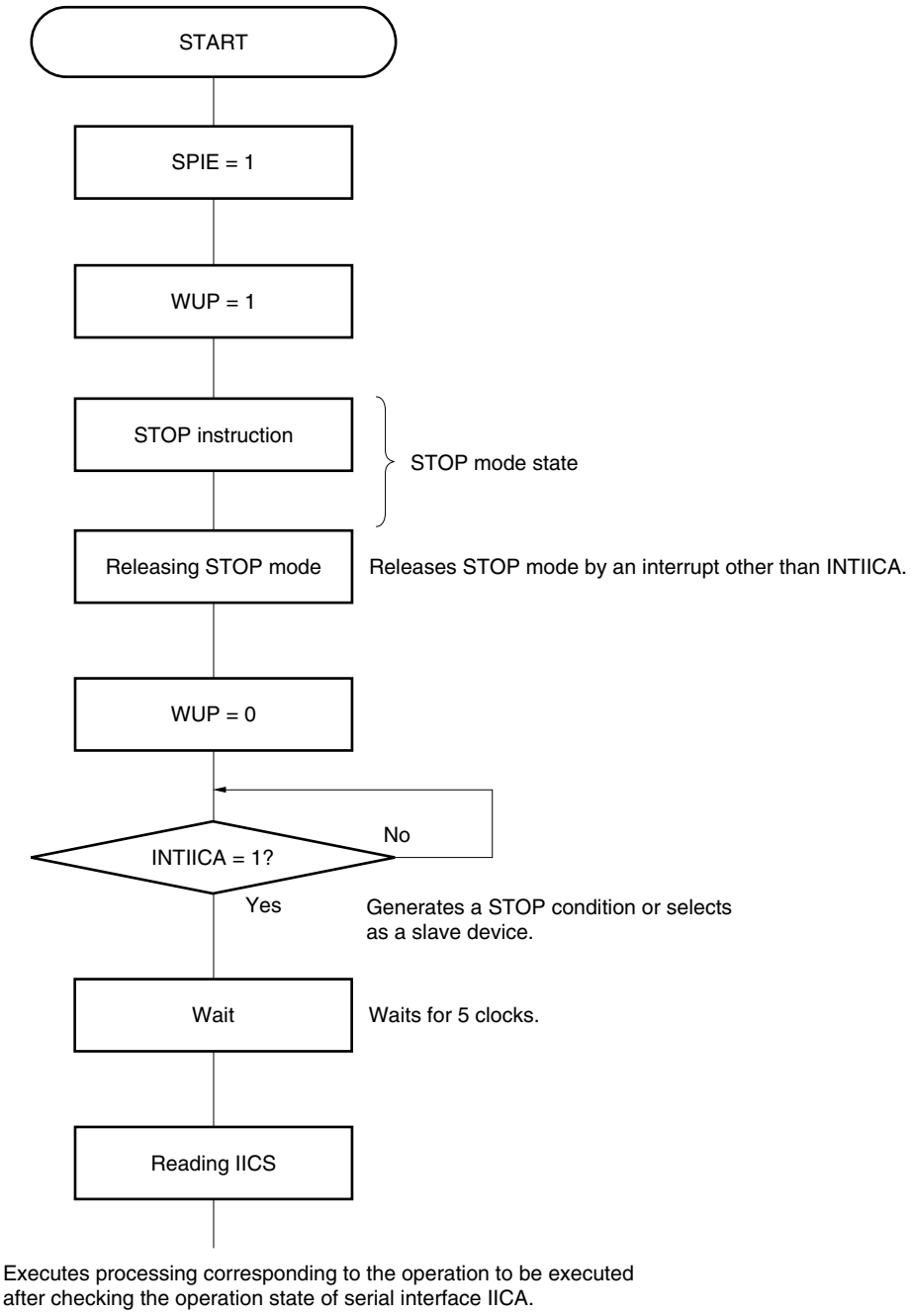


Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICA) generated from serial interface IICA.

- Master device operation: Flow shown in Figure 13-24
- Slave device operation: Same as the flow in Figure 13-23

Figure 13-24. When Operating as Master Device after Releasing STOP Mode other than by INTIICA



13.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IICA flag register (IICF) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL) of IICA control register 0 (IICCTL0) to 1 and saving communication).

If bit 1 (STT) of IICCTL0 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 (IICA) after bit 4 (SPIE) of IICCTL0 was set to 1, and it was detected by generation of an interrupt request signal (INTIICA) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to IICA before the stop condition is detected is invalid.
When STT 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

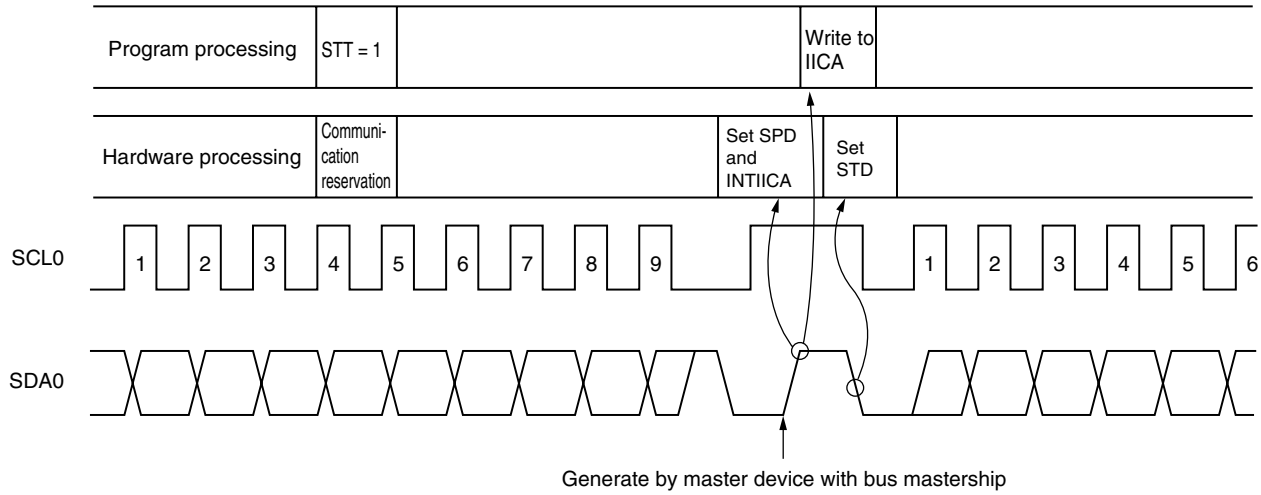
Check whether the communication reservation operates or not by using MSTS (bit 7 of the IICA status register (IICS)) after STT is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

Wait time from setting STT = 1 to checking the MSTS flag: (IICWL setting value + IICWH setting value + 4 clocks) / f_{CLK} + $t_F \times 2$

Remark IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

Figure 13-25 shows the communication reservation timing.

Figure 13-25. Communication Reservation Timing



Remark IICA: IICA shift register
STT: Bit 1 of IICA control register 0 (IICCTL0)
STD: Bit 1 of IICA status register (IICS)
SPD: Bit 0 of IICA status register (IICS)

Communication reservations are accepted via the timing shown in Figure 13-26. After bit 1 (STD) of the IICA status register (IICS) is set to 1, a communication reservation can be made by setting bit 1 (STT) of IICA control register 0 (IICCTL0) to 1 before a stop condition is detected.

Figure 13-26. Timing for Accepting Communication Reservations

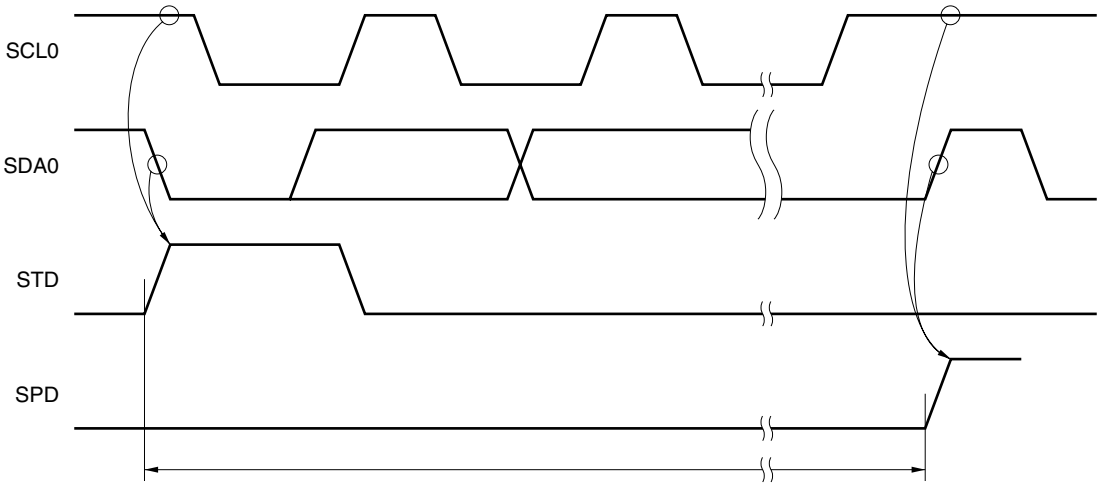
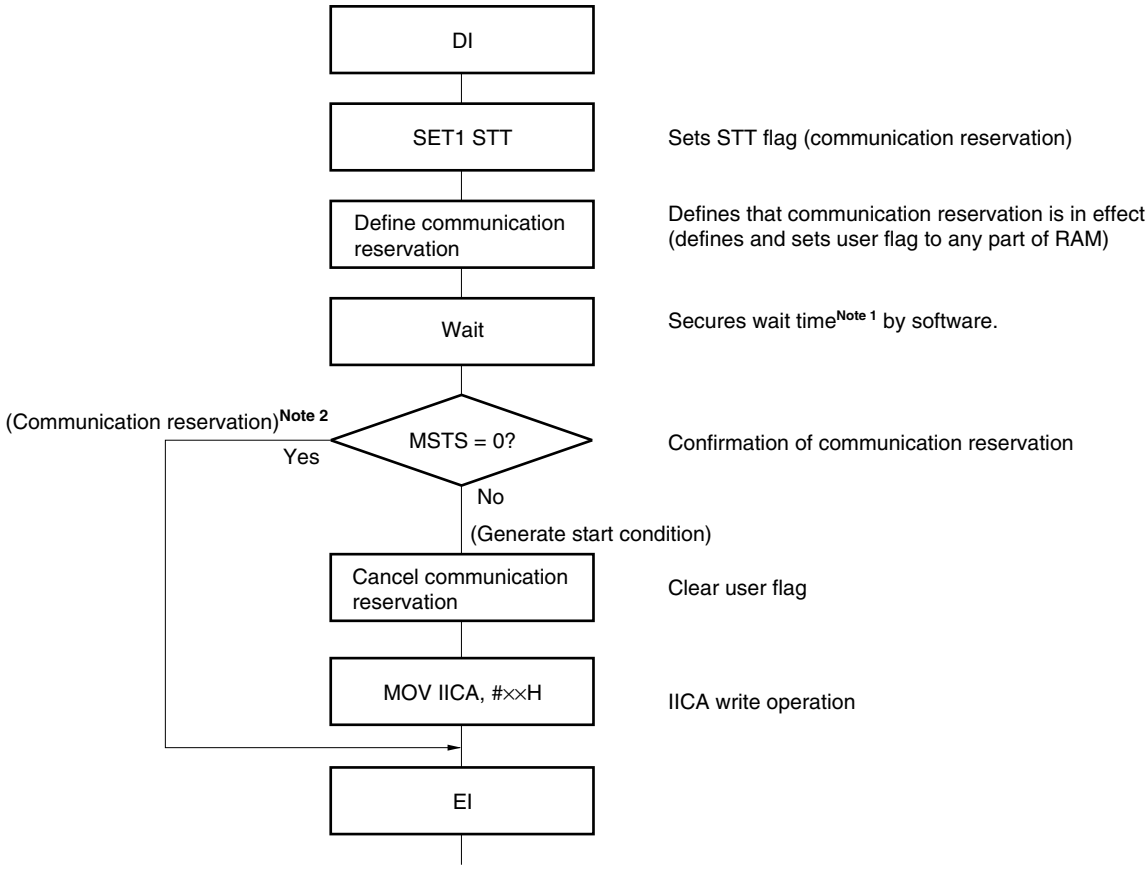


Figure 13-27 shows the communication reservation protocol.

Figure 13-27. Communication Reservation Protocol



Notes 1. The wait time is calculated as follows.
(IICWL setting value + IICWH setting value + 4 clocks) / f_{CLK} + $t_F \times 2$

2. The communication reservation operation executes a write to the IICA shift register (IICA) when a stop condition interrupt request occurs.

Remark STT: Bit 1 of IICA control register 0 (IICCTL0)
MSTS: Bit 7 of IICA status register (IICS)
IICA: IICA shift register
IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

(2) When communication reservation function is disabled (bit 0 (IICRSV) of IICA flag register (IICF) = 1)

When bit 1 (STT) of IICA control register 0 (IICCTL0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL) of IICCTL0 to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCF (bit 7 of IICF). It takes up to 5 clocks until STCF is set to 1 after setting STT = 1. Therefore, secure the time by software.

13.5.15 Cautions

(1) When STCEN = 0

Immediately after I^2C operation is enabled (IICE = 1), the bus communication status (IICBSY = 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 1 (IICCTL1).
<2> Set bit 7 (IICE) of IICA control register 0 (IICCTL0) to 1.
<3> Set bit 0 (SPT) of IICCTL0 to 1.

(2) When STCEN = 1

Immediately after I^2C operation is enabled (IICE = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I^2C communications are already in progress

If I^2C operation is enabled and the device participates in communication already in progress when the SDA0 pin is low and the SCL0 pin is high, the macro of I^2C recognizes that the SDA0 pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, $\overline{ACK}$ is returned, but this interferes with other I^2C communications. To avoid this, start I^2C in the following sequence.

<1> Clear bit 4 (SPIE) of IICCTL0 to 0 to disable generation of an interrupt request signal (INTIICA) when the stop condition is detected.
<2> Set bit 7 (IICE) of IICCTL0 to 1 to enable the operation of I^2C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL) of IICCTL0 to 1 before $\overline{ACK}$ is returned (4 to 80 clocks after setting IICE to 1), to forcibly disable detection.

(4) Setting STT and SPT (bits 1 and 0 of IICCTL0) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set SPIE (bit 4 of IICTL0) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to IICA 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 SPIE to 1 when MSTS (bit 7 of IICS) is detected by software.

13.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the 78K0R/Lx3 microcontrollers as the master in a single master system is shown below.
This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multimaster system

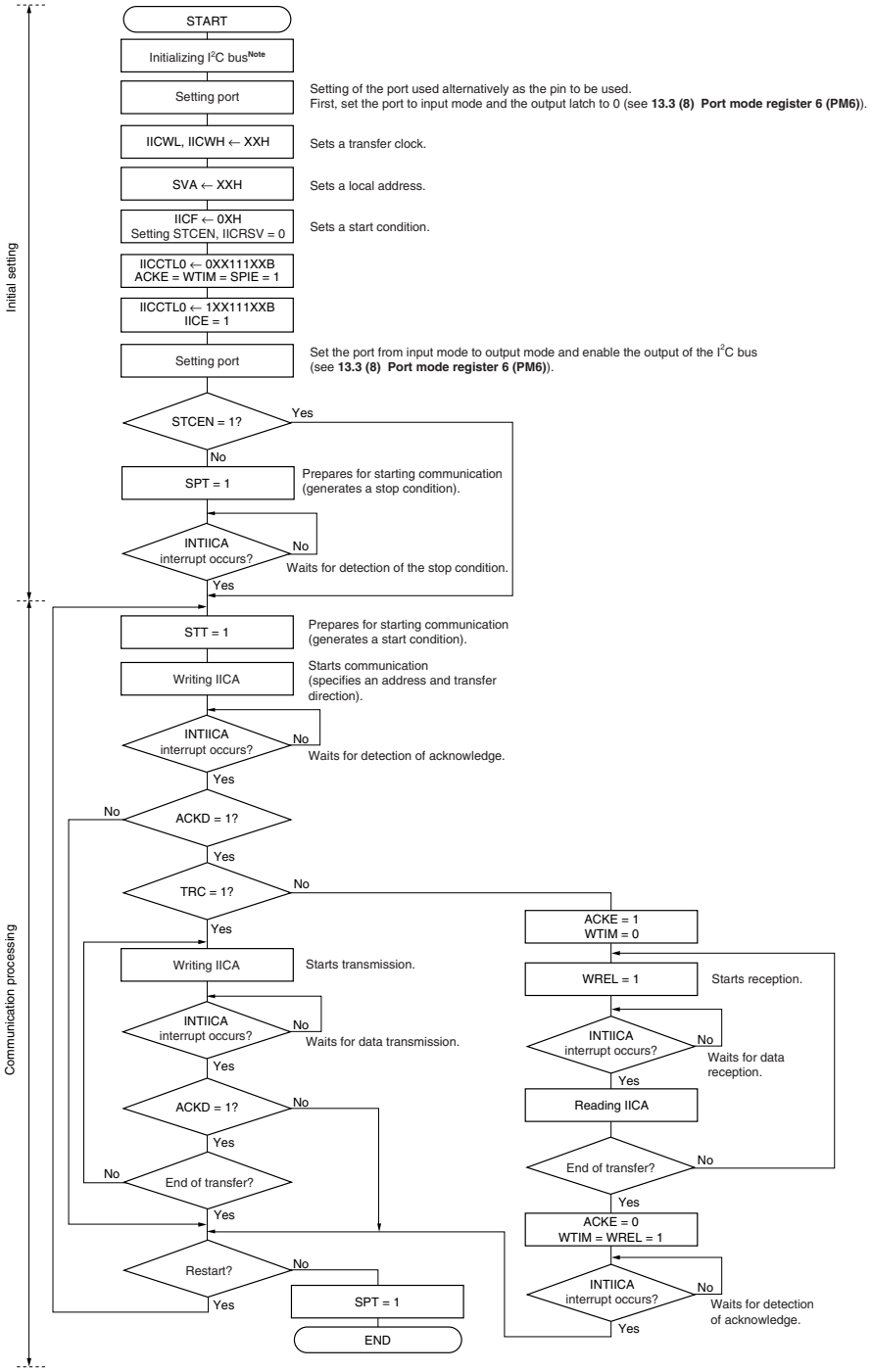
In the I^2C bus multimaster system, whether the bus is released or used cannot be judged by the I^2C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the 78K0R/Lx3 microcontrollers take part in a communication with bus released state.
This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the 78K0R/Lx3 microcontrollers loose in arbitration and are 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 78K0R/Lx3 microcontrollers are used as the I^2C bus slave is shown below.
When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICA interrupt occurrence (communication waiting). When an INTIICA interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.
By checking the flags, necessary communication processing is performed.

(1) Master operation in single-master system

Figure 13-28. Master Operation in Single-Master System

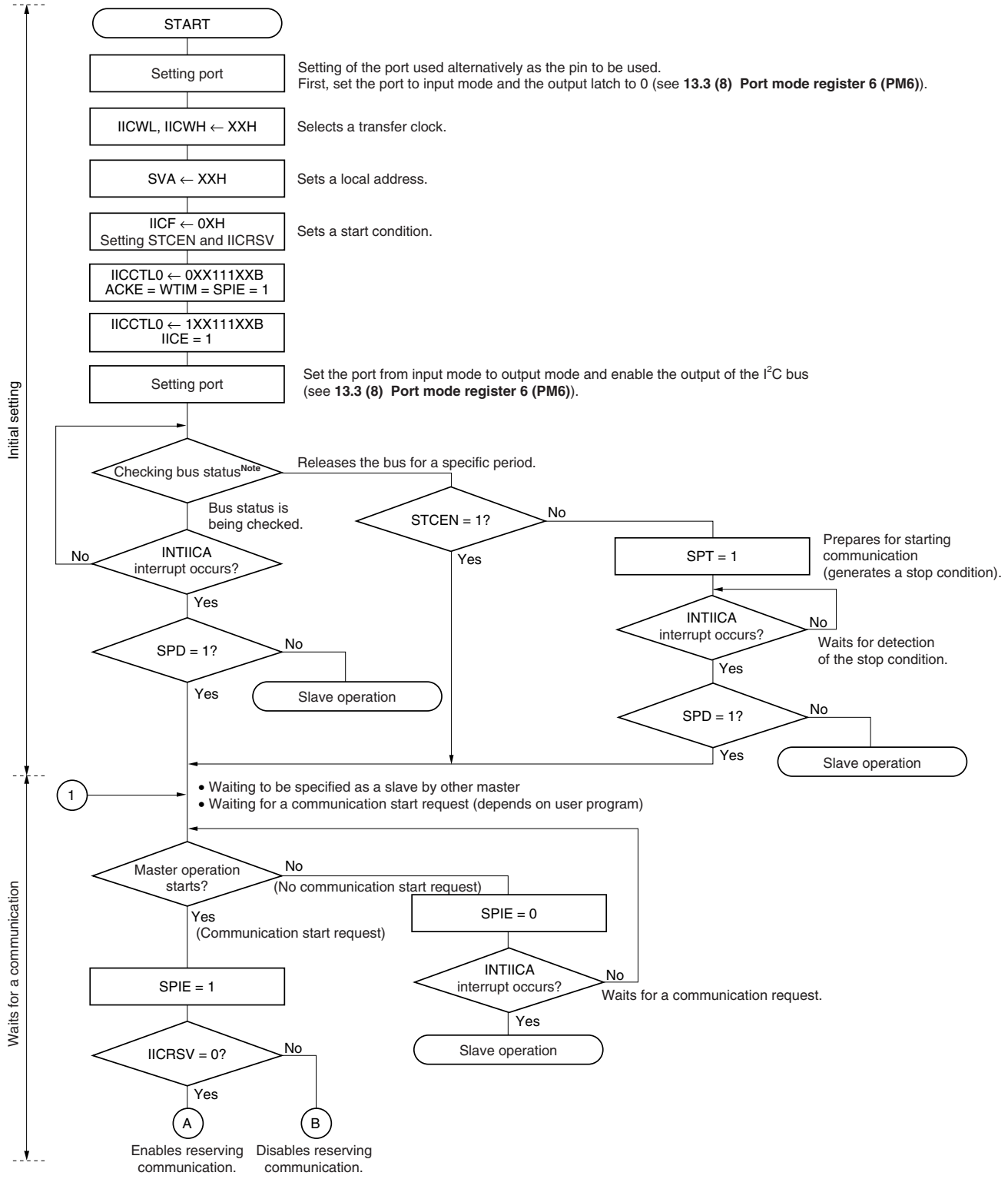


Note Release (SCL0 and SDA0 pins = high level) the I^2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDA0 pin, for example, set the SCL0 pin in the output port mode, and output a clock pulse from the output port until the SDA0 pin is constantly at high level.

Remark Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

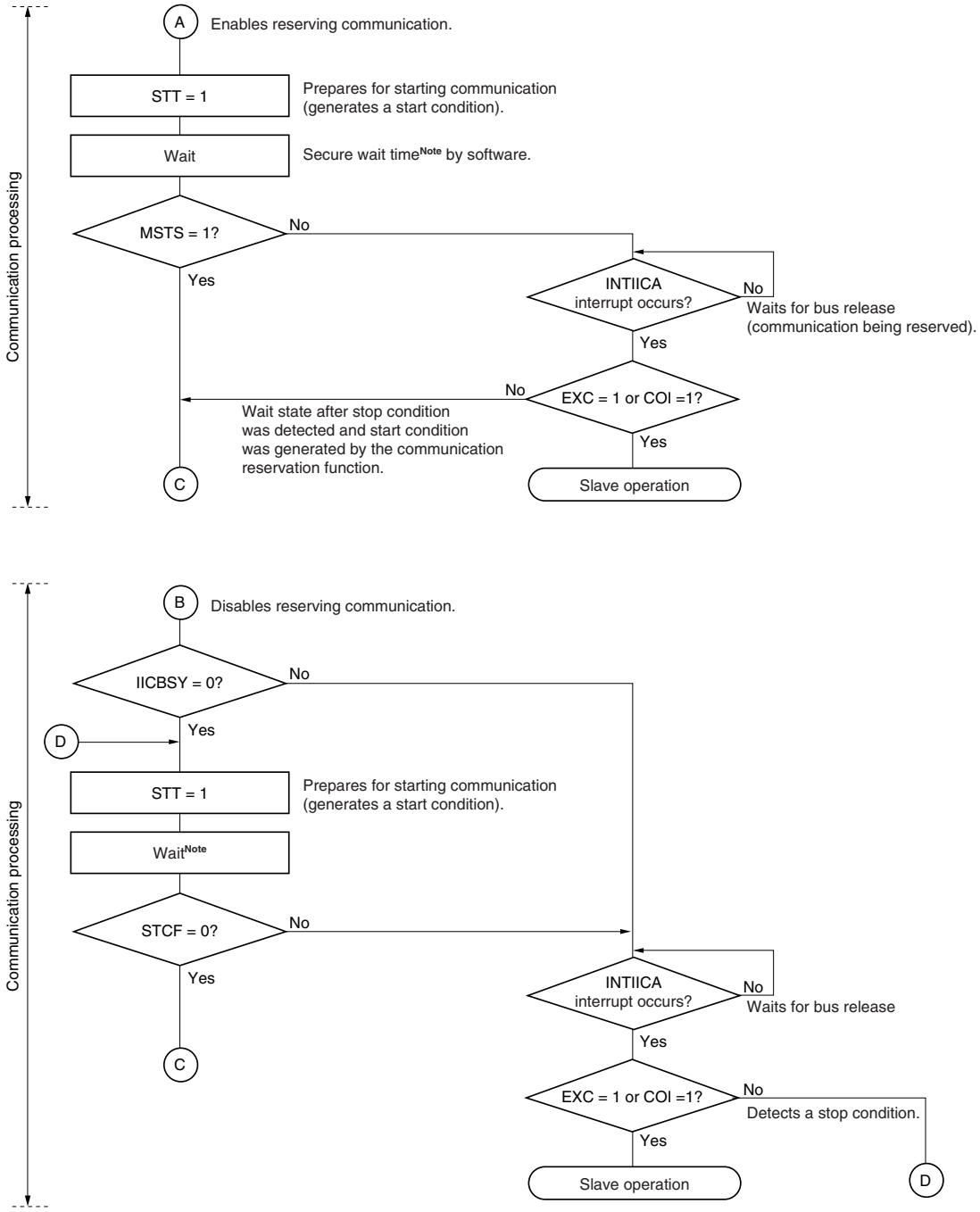
(2) Master operation in multi-master system

Figure 13-29. Master Operation in Multi-Master System (1/3)



Note Confirm that the bus is released (CLD bit = 1, DAD bit = 1) for a specific period (for example, for a period of one frame). If the SDA0 pin is constantly at low level, decide whether to release the I^2C bus (SCL0 and SDA0 pins = high level) in conformance with the specifications of the product that is communicating.

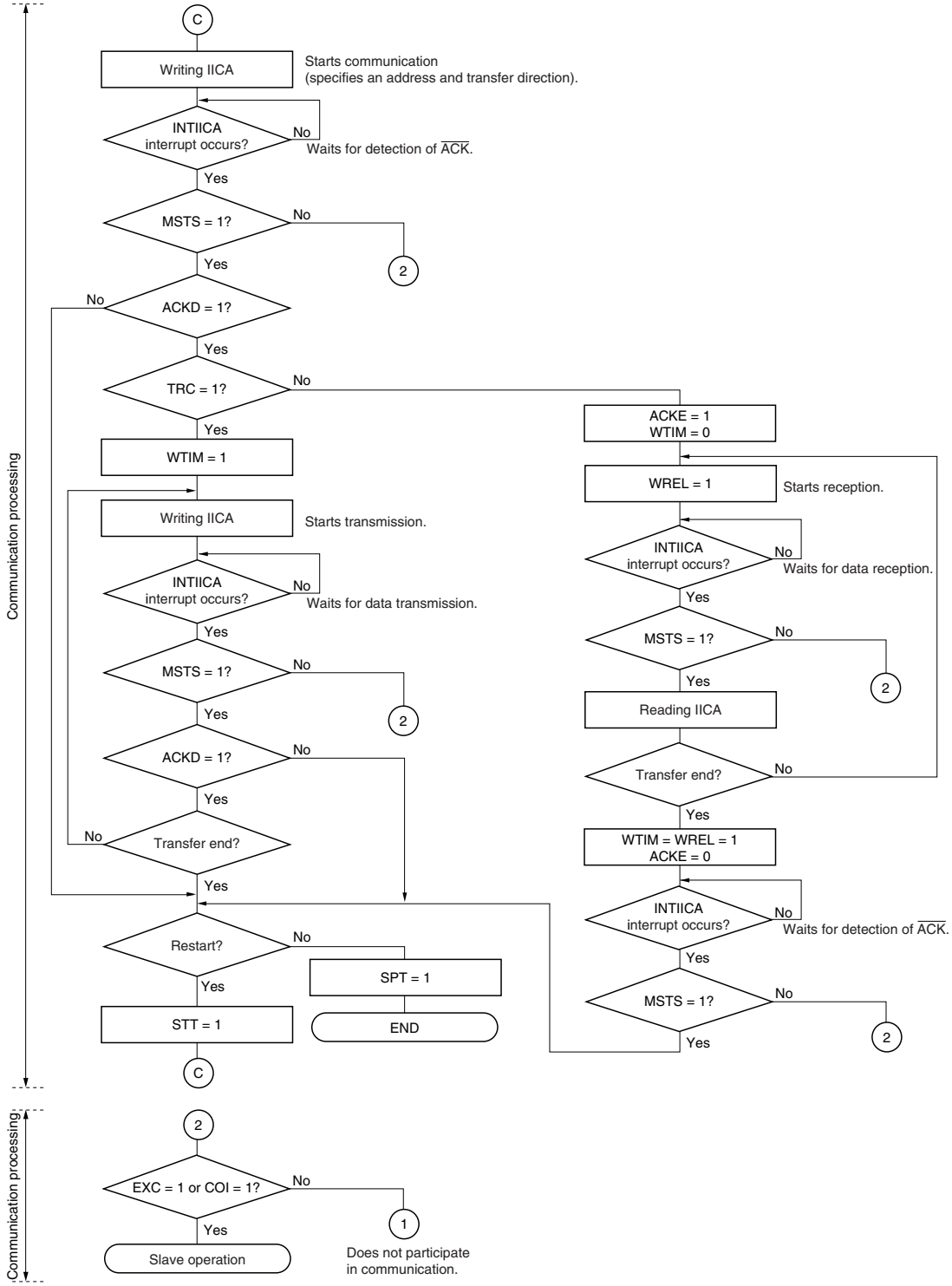
Figure 13-29. Master Operation in Multi-Master System (2/3)



Note The wait time is calculated as follows.
(IICWL setting value + IICWH setting value + 4 clocks) / f_{CLK} + $t_F \times 2$

Remark IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

Figure 13-29. Master Operation in Multi-Master System (3/3)



Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. To use the device as a master in a multi-master system, read the MSTS bit each time interrupt INTIICA has occurred to check the arbitration result.

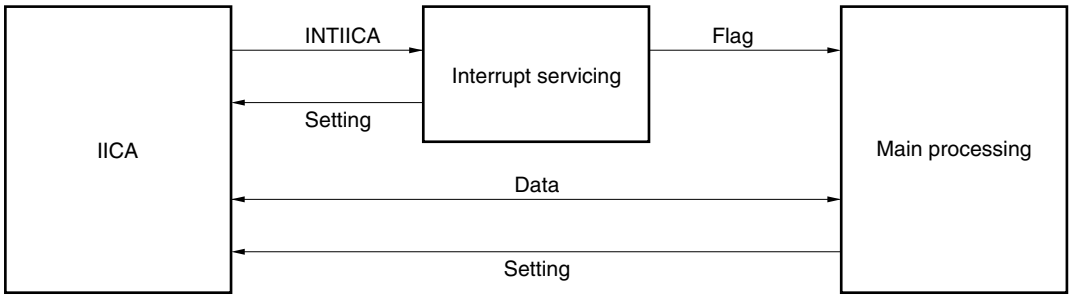
3. To use the device as a slave in a multi-master system, check the status by using the IICS and IICF registers each time interrupt INTIICA has occurred, and determine the processing to be performed next.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICA 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 INTIICA 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 INTIICA.

<1> Communication mode flag

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of $\overline{\text{ACK}}$ from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICA 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 TRC.

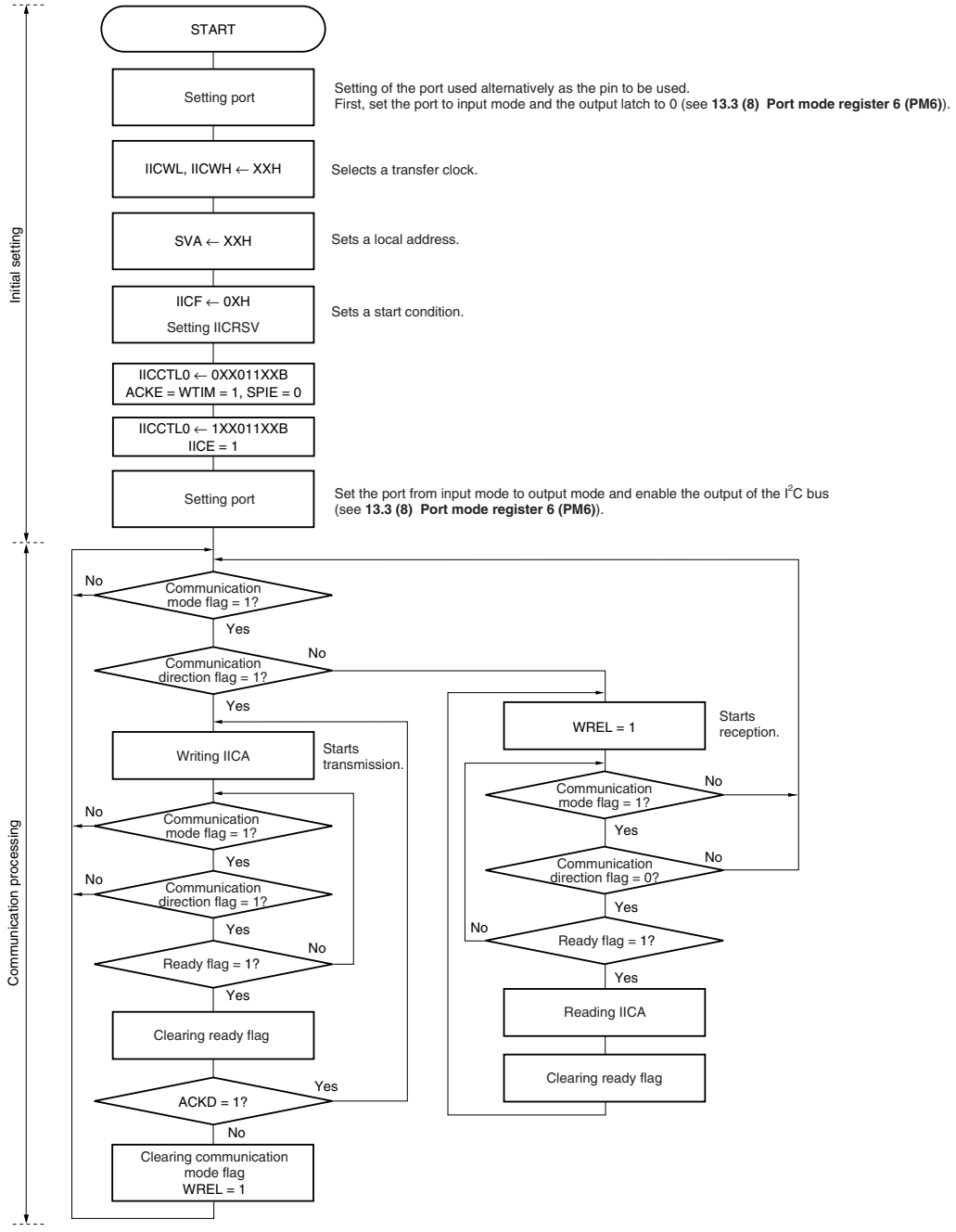
The main processing of the slave operation is explained next.

Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns $\overline{ACK}$. If $\overline{ACK}$ is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, $\overline{ACK}$ is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Figure 13-30. Slave Operation Flowchart (1)



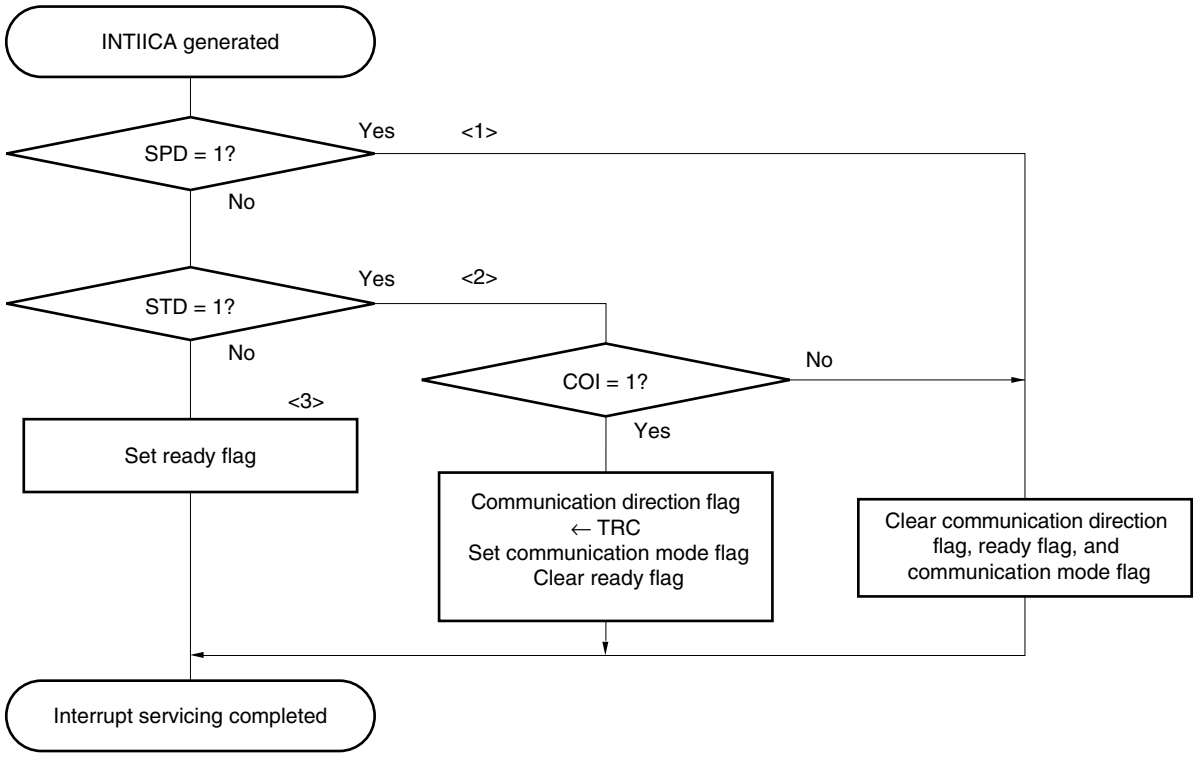
Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

An example of the processing procedure of the slave with the INTIICA interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICA interrupt checks the status, and the following operations are performed.

<1> Communication is stopped if the stop condition is issued.
<2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).
<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in Figure 13-31 Slave Operation Flowchart (2).

Figure 13-31. Slave Operation Flowchart (2)



13.5.17 Timing of I²C interrupt request (INTIICA) occurrence

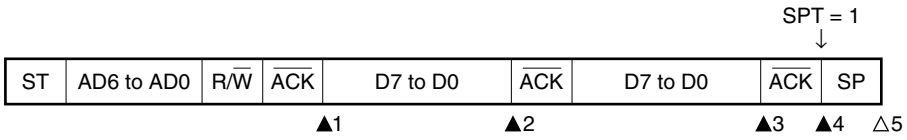
The timing of transmitting or receiving data and generation of interrupt request signal INTIICA, and the value of the IICS register when the INTIICA signal is generated are shown below.

Remark ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM = 0

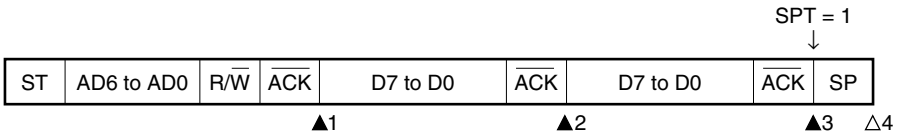


▲1: IICS = 1000×110B
▲2: IICS = 1000×000B
▲3: IICS = 1000×000B (Sets WTIM to 1)Note
▲4: IICS = 1000××00B (Sets SPT to 1)Note
△5: IICS = 00000001B

Note To generate a stop condition, set WTIM to 1 and change the timing for generating the INTIICA interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

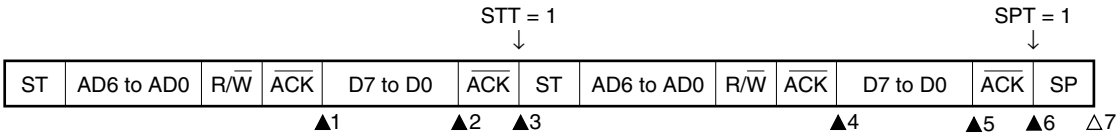


▲1: IICS = 1000×110B
▲2: IICS = 1000×100B
▲3: IICS = 1000××00B (Sets SPT to 1)
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM = 0

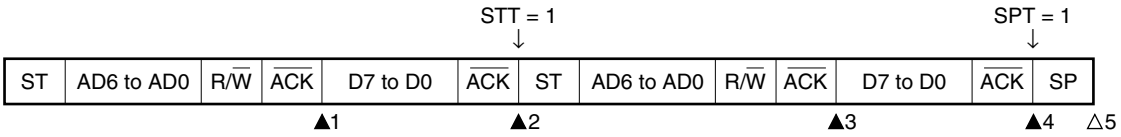


▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM to 1)[Note 1]
▲3: IICS = 1000××00B (Clears WTIM to 0[Note 2], sets STT to 1)
▲4: IICS = 1000×110B
▲5: IICS = 1000×000B (Sets WTIM to 1)[Note 3]
▲6: IICS = 1000××00B (Sets SPT to 1)
△7: IICS = 00000001B

Notes 1. To generate a start condition, set WTIM to 1 and change the timing for generating the INTIICA interrupt request signal.
2. Clear WTIM to 0 to restore the original setting.
3. To generate a stop condition, set WTIM to 1 and change the timing for generating the INTIICA interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

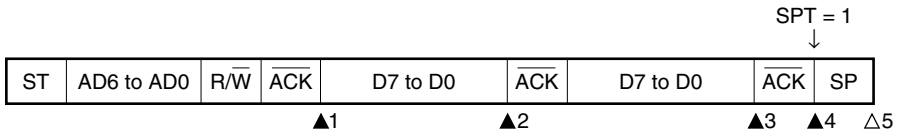


▲1: IICS = 1000×110B
▲2: IICS = 1000××00B (Sets STT to 1)
▲3: IICS = 1000×110B
▲4: IICS = 1000××00B (Sets SPT to 1)
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM = 0

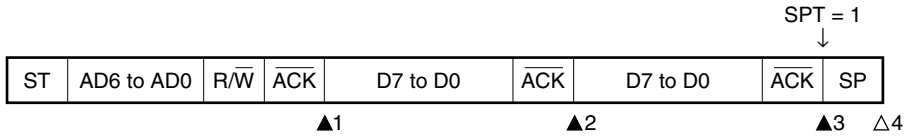


▲1: IICS = 1010×110B
▲2: IICS = 1010×000B
▲3: IICS = 1010×000B (Sets WTIM to 1)[Note]
▲4: IICS = 1010××00B (Sets SPT to 1)
△5: IICS = 00000001B

Note To generate a stop condition, set WTIM to 1 and change the timing for generating the INTIICA interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



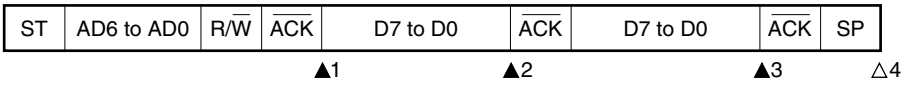
▲1: IICS = 1010×110B
▲2: IICS = 1010×100B
▲3: IICS = 1010××00B (Sets SPT to 1)
△4: IICS = 00001001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

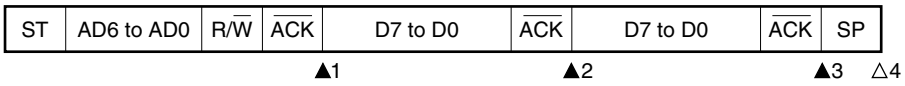
(i) When WTIM = 0



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

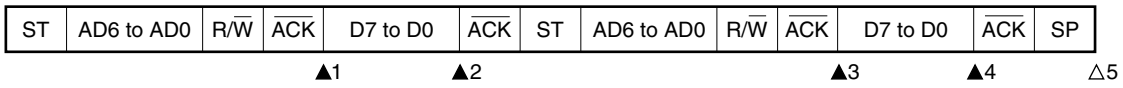


▲1: IICS = 0001×110B
▲2: IICS = 0001×100B
▲3: IICS = 0001××00B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

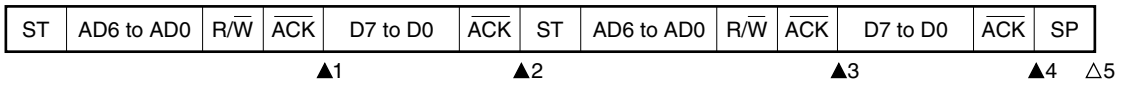
(i) When WTIM = 0 (after restart, matches with SVA)



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×110B
▲4: IICS = 0001×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, matches with SVA)

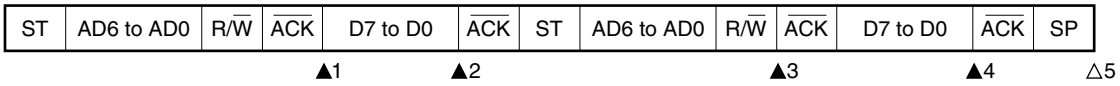


▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 0001×110B
▲4: IICS = 0001××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

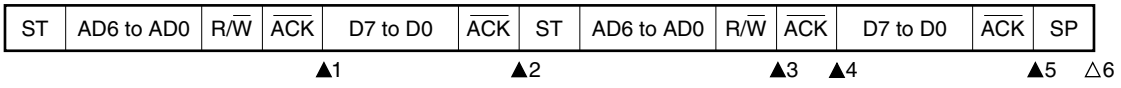
(i) When WTIM = 0 (after restart, does not match address (= extension code))



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0010×010B
▲4: IICS = 0010×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, does not match address (= extension code))

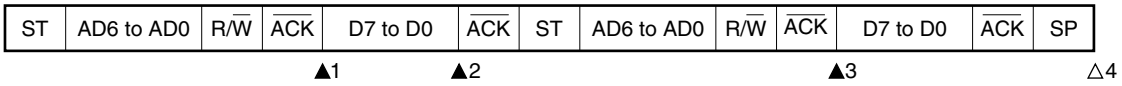


▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 0010×010B
▲4: IICS = 0010×110B
▲5: IICS = 0010××00B
△6: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

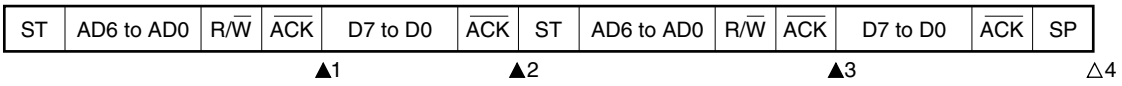
(i) When WTIM = 0 (after restart, does not match address (= not extension code))



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 00000110B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, does not match address (= not extension code))



▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 00000110B
△4: IICS = 00000001B

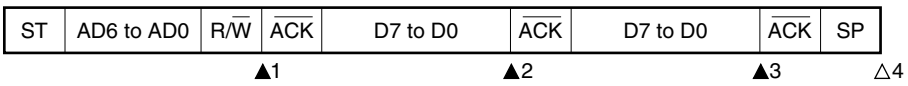
Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(3) Slave device operation (when receiving extension code)

The device is always participating in communication when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

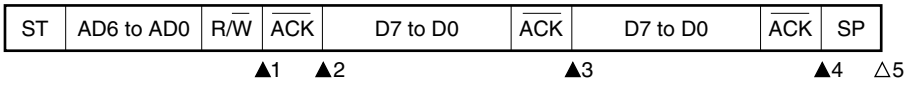
(i) When WTIM = 0



▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0010×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

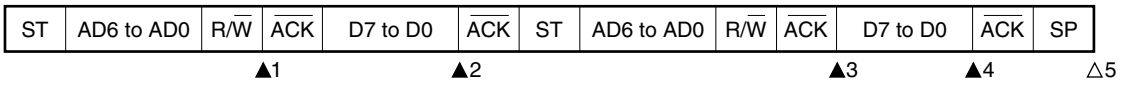


▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010×100B
▲4: IICS = 0010××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

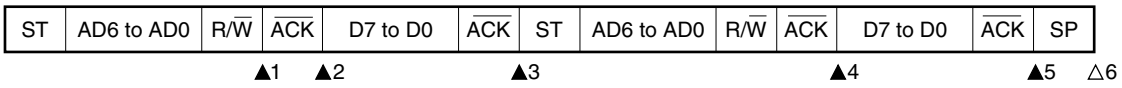
(i) When WTIM = 0 (after restart, matches SVA)



▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0001×110B
▲4: IICS = 0001×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, matches SVA)

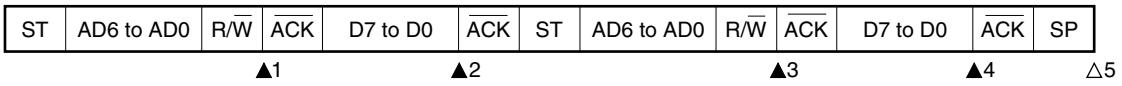


▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010××00B
▲4: IICS = 0001×110B
▲5: IICS = 0001××00B
△6: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

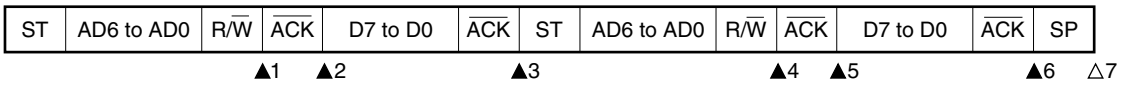
(i) When WTIM = 0 (after restart, extension code reception)



▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0010×010B
▲4: IICS = 0010×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, extension code reception)



▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010××00B
▲4: IICS = 0010×010B
▲5: IICS = 0010×110B
▲6: IICS = 0010××00B
△7: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

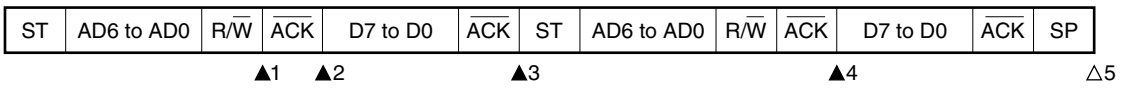
(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM = 0 (after restart, does not match address (= not extension code))

▲1: IICS = 00100010B
▲2: IICS = 00100000B
▲3: IICS = 00000110B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, does not match address (= not extension code))

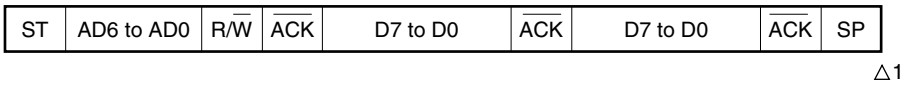


▲1: IICS = 00100010B
▲2: IICS = 00100110B
▲3: IICS = 00100×00B
▲4: IICS = 00000110B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICS = 00000001B

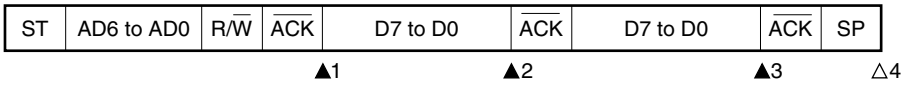
Remark △: Generated only when SPIE = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS bit each time interrupt request signal INTIICA has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

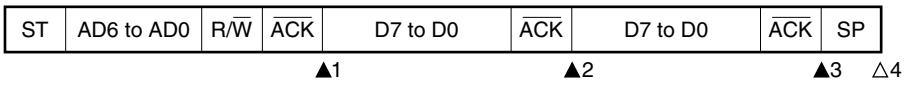
(i) When WTIM = 0



▲1: IICS = 0101×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

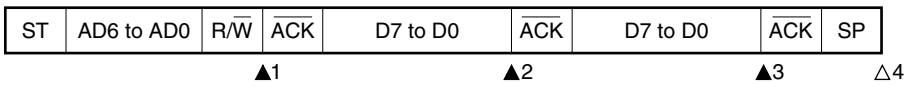


▲1: IICS = 0101×110B
▲2: IICS = 0001×100B
▲3: IICS = 0001××00B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

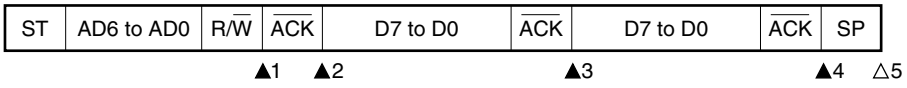
(i) When WTIM = 0



▲1: IICS = 0110×010B
▲2: IICS = 0010×000B
▲3: IICS = 0010×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



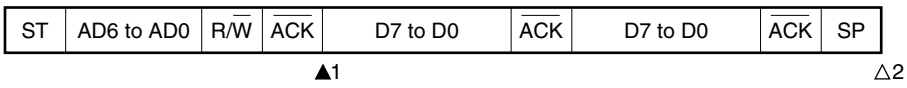
▲1: IICS = 0110×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010×100B
▲4: IICS = 0010××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS bit each time interrupt request signal INTIICA has occurred to check the arbitration result.

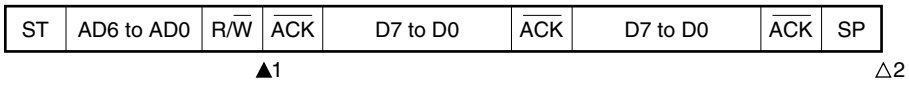
(a) When arbitration loss occurs during transmission of slave address data (when WTIM = 1)



▲1: IICS = 01000110B
△2: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1

(b) When arbitration loss occurs during transmission of extension code

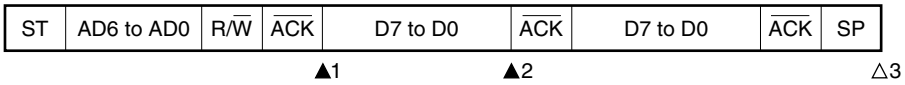


▲1: IICS = 0110×010B
Sets LREL = 1 by software
△2: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

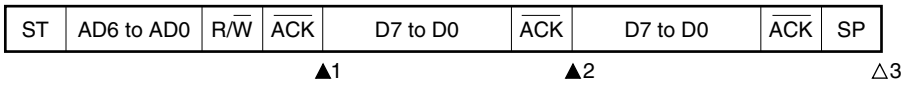
(i) When WTIM = 0



▲1: IICS = 10001110B
▲2: IICS = 01000000B
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1

(ii) When WTIM = 1

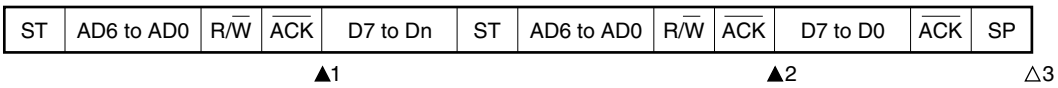


▲1: IICS = 10001110B
▲2: IICS = 01000100B
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1

(d) When loss occurs due to restart condition during data transfer

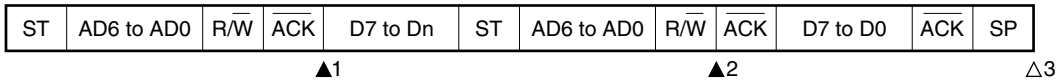
(i) Not extension code (Example: unmatches with SVA)



▲1: IICS = 1000×110B
▲2: IICS = 01000110B
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care
n = 6 to 0

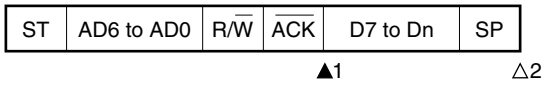
(ii) Extension code



▲1: IICS = 1000×110B
▲2: IICS = 01100010B
Sets LREL = 1 by software
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care
n = 6 to 0

(e) When loss occurs due to stop condition during data transfer

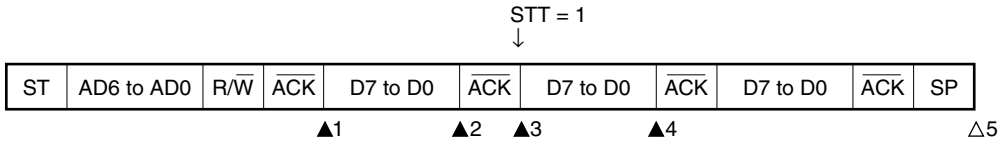


▲1: IICS = 10000110B
△2: IICS = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care
n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

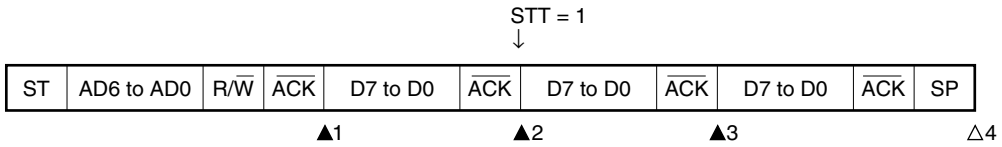
(i) When WTIM = 0



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM to 1)
▲3: IICS = 1000×100B (Clears WTIM to 0)
▲4: IICS = 01000000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

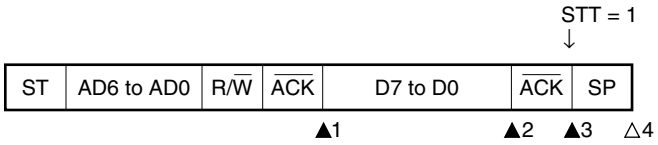


▲1: IICS = 1000×110B
▲2: IICS = 1000×100B (Sets STT to 1)
▲3: IICS = 01000100B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

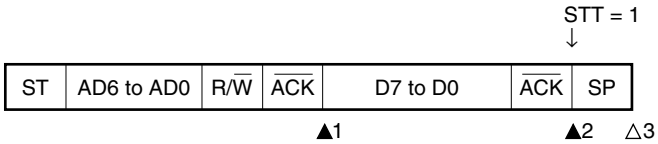
(i) When WTIM = 0



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM to 1)
▲3: IICS = 1000××00B (Sets STT to 1)
△4: IICS = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

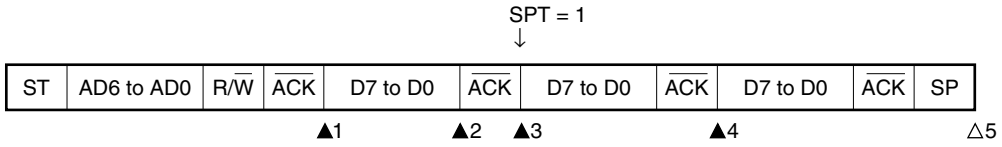


▲1: IICS = 1000×110B
▲2: IICS = 1000××00B (Sets STT to 1)
△3: IICS = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

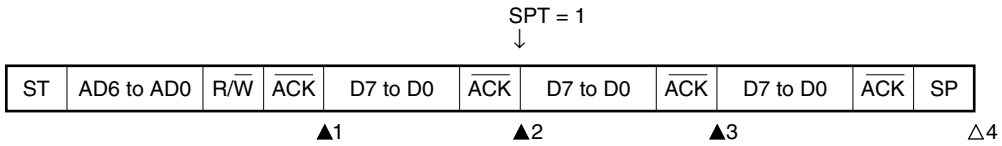
(i) When WTIM = 0



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM to 1)
▲3: IICS = 1000×100B (Clears WTIM to 0)
▲4: IICS = 01000100B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



▲1: IICS = 1000×110B
▲2: IICS = 1000×100B (Sets SPT to 1)
▲3: IICS = 01000100B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

13.6 Timing Charts

When using the I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC bit (bit 3 of the IICA status register (IICS)), which specifies the data transfer direction, and then starts serial communication with the slave device.

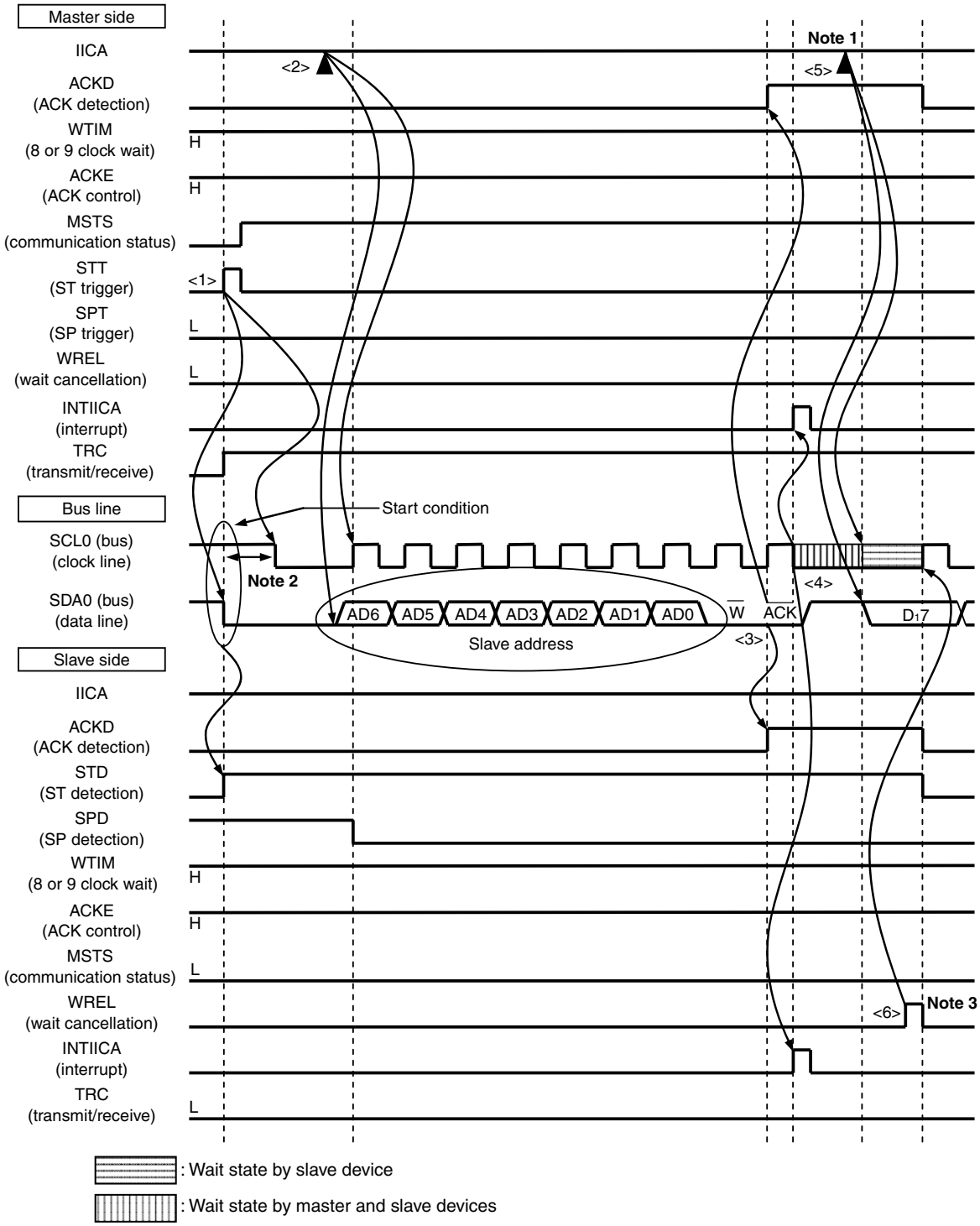
Figures 13-32 and 13-33 show timing charts of the data communication.

The IICA shift register (IICA)'s shift operation is synchronized with the falling edge of the serial clock (SCL0). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA0 pin.

Data input via the SDA0 pin is captured into IICA at the rising edge of SCL0.

Figure 13-32. Example of Master to Slave Communication (When 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 IICA, not setting the WREL bit, in order to cancel a wait state during master transmission.

2. Make sure that the time between the fall of the SDA0 pin signal and the fall of the SCL0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 13-32 are explained below.

<1> The start condition trigger is set by the master device (STT = 1) and a start condition (SDA0 = 0 and SCL0 = 1) is generated once the bus data line goes low (SDA0 = 0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS = 1). The master device is ready to communicate once the bus clock line goes low (SCL0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register (IICA) and transmits the slave address.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The master device writes the data to transmit to the IICA register and releases the wait status that it set by the master device.

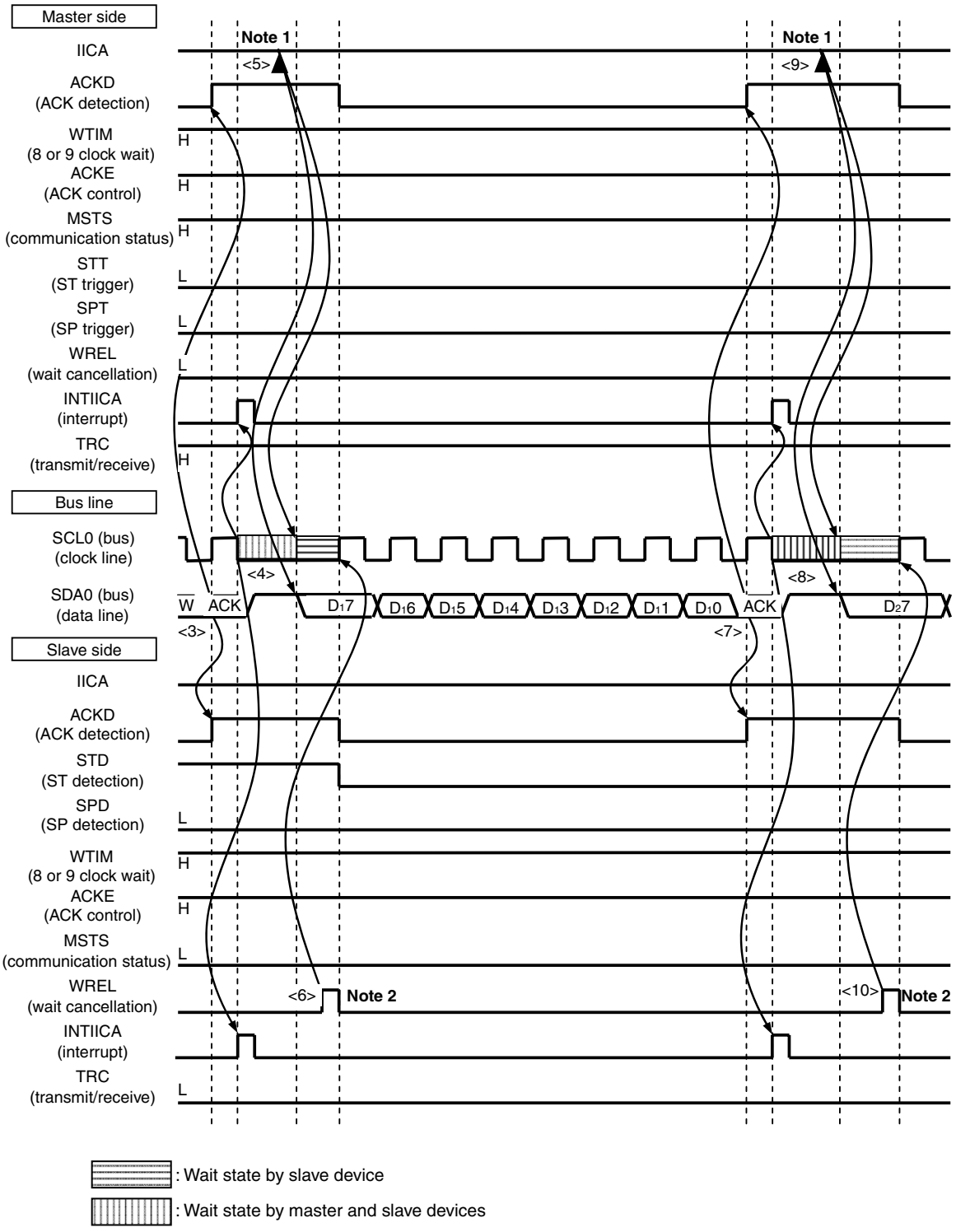
<6> If the slave device releases the wait status (WREL = 1), the master device starts transferring data to the slave device.

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: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 13-32 represent the entire procedure for communicating data using the I^2C bus. Figure 13-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 13-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 13-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 13-32. Example of Master to Slave Communication
(When 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 IICA, not setting the WREL bit, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 13-32 are explained below.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The master device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the master device.

<6> If the slave device releases the wait status (WREL = 1), the master device starts transferring data to the slave device.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<9> The master device writes the data to transmit to the IICA register and releases the wait status that it set by the master device.

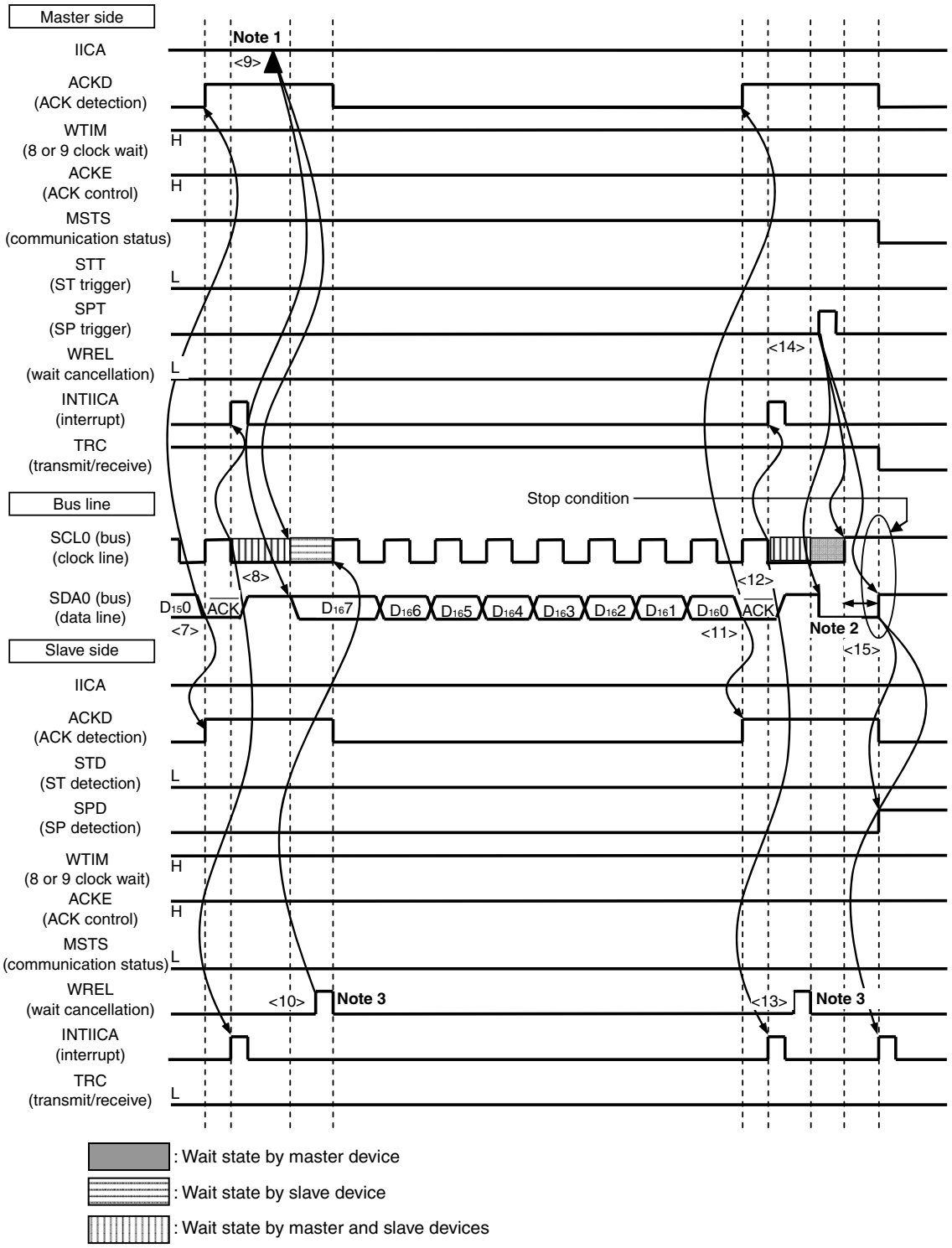
<10> The slave device reads the received data and releases the wait status (WREL = 1). The master device then starts transferring data to the slave device.

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: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 13-32 represent the entire procedure for communicating data using the I^2C bus. Figure 13-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 13-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 13-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 13-32. Example of Master to Slave Communication (When 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 IICA, not setting the WREL bit, in order to cancel a wait state during master transmission.

2. Make sure that the time between the rise of the SCL0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 13-32 are explained below.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register (IICA) 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 (WREL = 1). The master device then starts transferring data to the slave device.

<11> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<12> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<13> The slave device reads the received data and releases the wait status (WREL = 1).

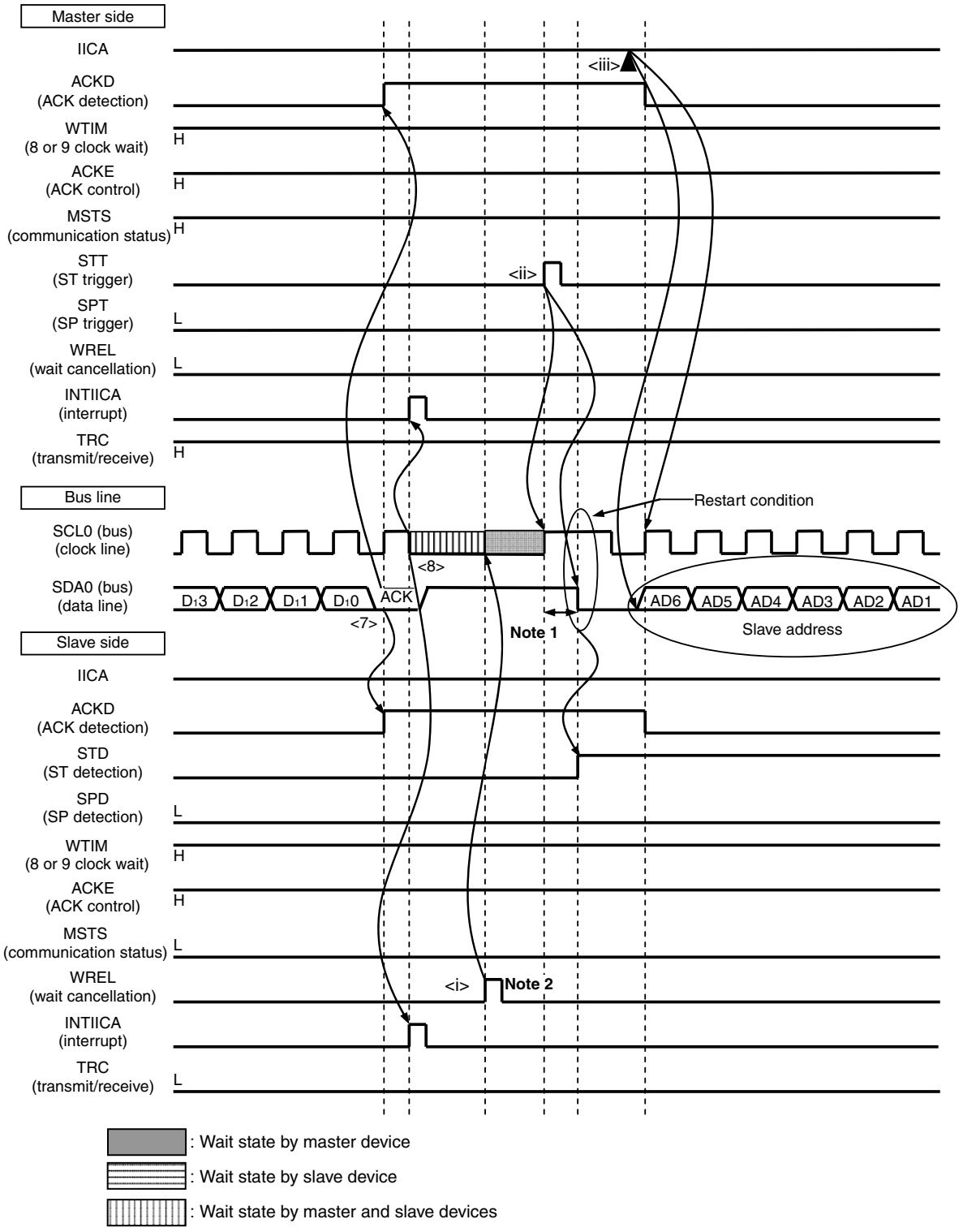
<14> After a stop condition trigger is set, the bus data line is cleared (SDA0 = 0) and the bus clock line is set (SCL0 = 1). The stop condition is then generated by setting the bus data line (SDA0 = 1) after the stop condition setup time has elapsed.

<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICA: stop condition).

Remark <1> to <15> in Figure 13-32 represent the entire procedure for communicating data using the I^2C bus. Figure 13-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 13-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 13-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 13-32. Example of Master to Slave Communication (When 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 SCL0 pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

2. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The following describes the operations in Figure 13-32 (4) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <1> to <3> are performed. These steps return the processing to step <3>, the data transmission step.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

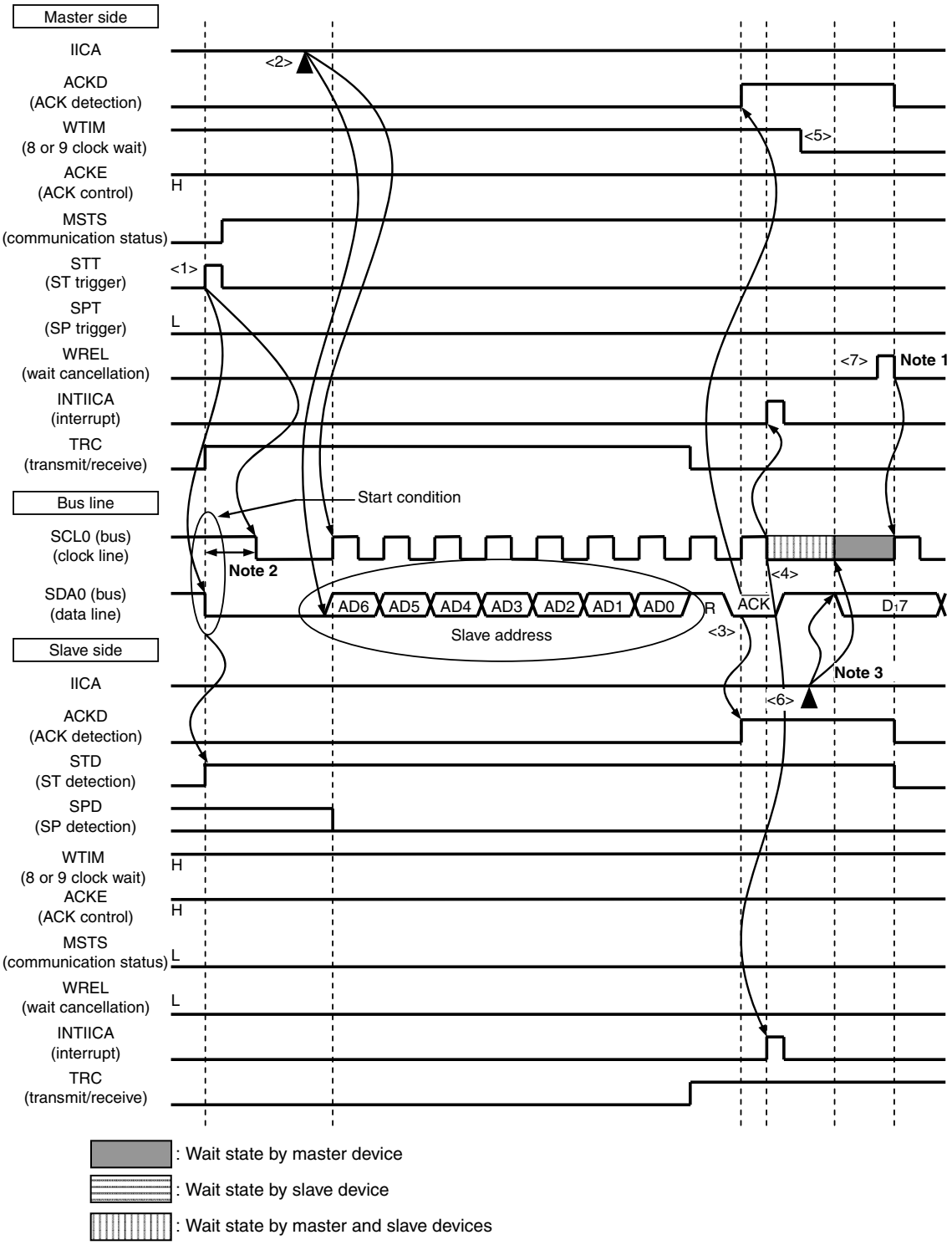
<i> The slave device reads the received data and releases the wait status (WREL = 1).

<ii> The start condition trigger is set again by the master device (STT = 1) and a start condition (SDA0 = 0 and SCL0 = 1) is generated once the bus clock line goes high (SCL0 = 1) and the bus data line goes low (SDA0 = 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 (SCL0 = 0) after the hold time has elapsed.

<iii> The master device writes the address + R/W (transmission) to the IICA shift register (IICA) and transmits the slave address.

Figure 13-33. Example of Slave to Master Communication (When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)

(1) Start condition ~ address ~ data



Notes 1. To cancel master wait, write “FFH” to IICA or set the WREL bit.

2. Make sure that the time between the fall of the SDA0 pin signal and the fall of the SCL0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during slave transmission.

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 13-33 are explained below.

<1> The start condition trigger is set by the master device (STT = 1) and a start condition (SDA0 = 0 and SCL0 = 1) is generated once the bus data line goes low (SDA0 = 0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS = 1). The master device is ready to communicate once the bus clock line goes low (SCL0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register (IICA) and transmits the slave address.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIM = 0).

<6> The slave device writes the data to transmit to the IICA register and releases the wait status that it set by the slave device.

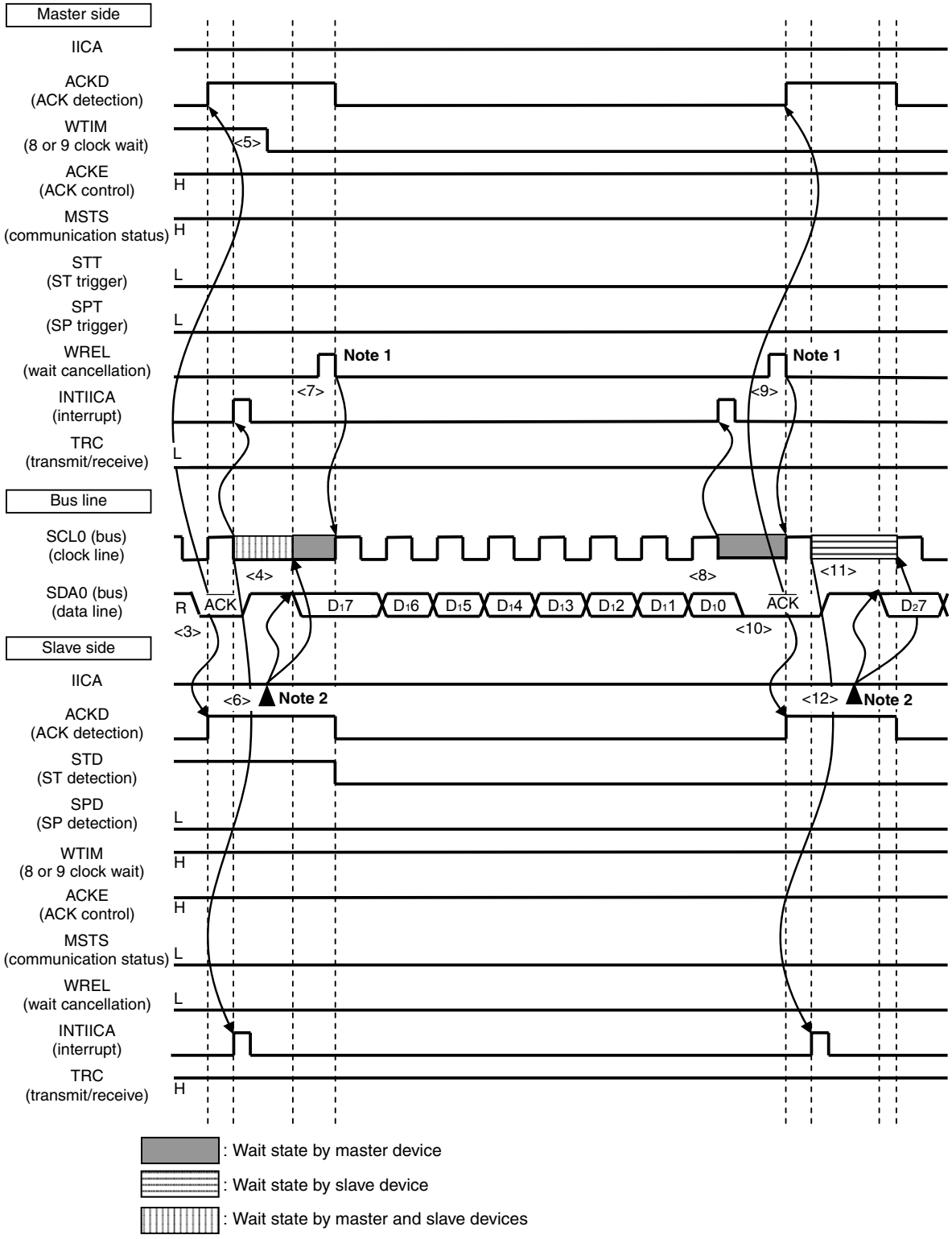
<7> If the master device releases the wait status (WREL = 1), the slave device starts transferring data to the master device.

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: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 13-33 represent the entire procedure for communicating data using the I^2C bus. Figure 13-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 13-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 13-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 13-33. Example of Slave to Master Communication (When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)

(2) Address ~ data ~ data



Notes 1. To cancel master wait, write “FFH” to IICA or set the WREL bit.

2. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during slave transmission.

The meanings of <3> to <12> in (2) Address ~ data ~ data in Figure 13-33 are explained below.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIM = 0).

<6> The slave device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the slave device.

<7> If the master device releases the wait status (WREL = 1), the slave device starts transferring data to the master device.

<8> The master device sets a wait status (SCL0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA: end of transfer). 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 (WREL = 1).

<10> The ACK is detected by the slave device (ACKD = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA: end of transfer).

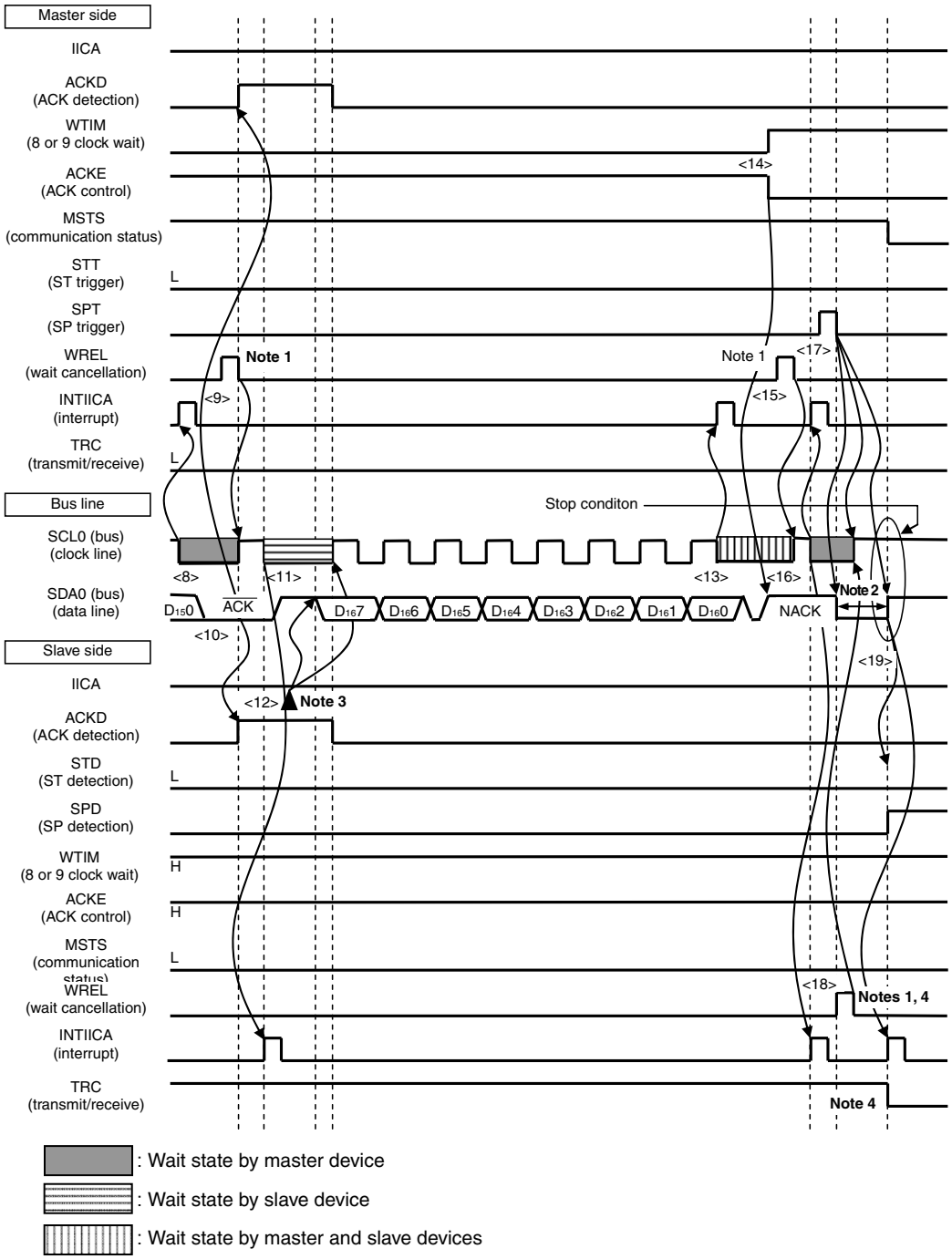
<12> The slave device writes the data to transmit to the IICA register and releases the wait status that it set by the slave device. The slave device then starts transferring data to the master device.

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: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 13-33 represent the entire procedure for communicating data using the I^2C bus. Figure 13-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 13-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 13-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 13-33. Example of Slave to Master Communication (When 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 1. To cancel a wait state, write “FFH” to IICA or set the WREL bit.

2. Make sure that the time between the rise of the SCL0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during slave transmission.

4. If a wait state during slave transmission is canceled by setting the WREL bit, the TRC bit will be cleared.

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 13-33 are explained below.

<8> The master device sets a wait status (SCL0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA: end of transfer). 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 (WREL = 1).

<10> The ACK is detected by the slave device (ACKD = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA: end of transfer).

<12> The slave device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the slave device. The slave device then starts transferring data to the master device.

<13> The master device issues an interrupt (INTIICA: end of transfer) at the falling edge of the 8th clock, and sets a wait status (SCL0 = 0). Because ACK control (ACKE = 1) is performed, the bus data line is at the low level (SDA0 = 0) at this stage.

<14> The master device sets NACK as the response (ACKE = 0) and changes the timing at which it sets the wait status to the 9th clock.

<15> If the master device releases the wait status (WREL = 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 (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<17> When the master device issues a stop condition (SPT = 1), the bus data line is cleared (SDA0 = 0) and the master device releases the wait status. The master device then waits until the bus clock line is set (SCL0 = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the wait status (WREL = 1) to end communication. Once the slave device releases the wait status, the bus clock line is set (SCL0 = 1).

<19> Once the master device recognizes that the bus clock line is set (SCL0 = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDA0 = 1) and issues a stop condition. The slave device detects the generated stop condition and both the master device and slave device issue an interrupt (INTIICA: stop condition).

Remark <1> to <19> in Figure 13-33 represent the entire procedure for communicating data using the I^2C bus. Figure 13-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 13-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 13-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

CHAPTER 14 EXTENDED SFR (3rd SFR) INTERFACE

14.1 Functions of Extended SFR (3rd SFR) Interface

The extended SFR (3rd SFR) interface is implemented using a bank selection signal line and clocked serial communication. The extended SFR (3rd SFR) can be accessed by combining the bank selection signal line with serial communication. CSI10 that is assigned to channel 2 of serial array unit 0 is used for serial communication.

Extended SFR (3rd SFR) interface has the following two modes.

(1) Operation stop mode

This mode is used when extended SFR (3rd SFR) interface communication is not performed and can enable a reduction in the power consumption. For details, see **14.4.1 Operation stop mode**.

(2) Serial I/O mode

In this mode, 8-bit data communication is performed by using two chip select signals (internal signals: P30 and P31), a serial clock (internal signal: $\overline{\text{SCK10}}$), and serial data (internal signals: SI10 and SO10).

For details, see **14.4.2 Serial I/O mode**.

Caution When using the extended SFR (3rd SFR) interface, be sure to make settings as follows.

- **Communication data length: 8 bits**
- **MSB-first data communication**
- **Master mode**
- **Set the phase selection of data and clock DAP02, CKP02 = 00**
- **While accessing to the related registers of extended SFR (3rd SFR) interface, execute DI command (IE bit of PSW = 0).**
- <R> **Transfer clock frequency for the extended SFR (3rd SFR) interface must be 1/4 or less of the clock frequency supplied to the interface, and must satisfy the following.**
  - **When $V_{DD} = LV_{DD} \geq 2.7$ V : Set transfer clock frequency to 1.25 MHz or less**
  - **When $V_{DD} = LV_{DD} < 2.7$ V : Set transfer clock frequency to 555 kHz or less**

14.2 Configuration of Extended SFR (3rd SFR) Interface

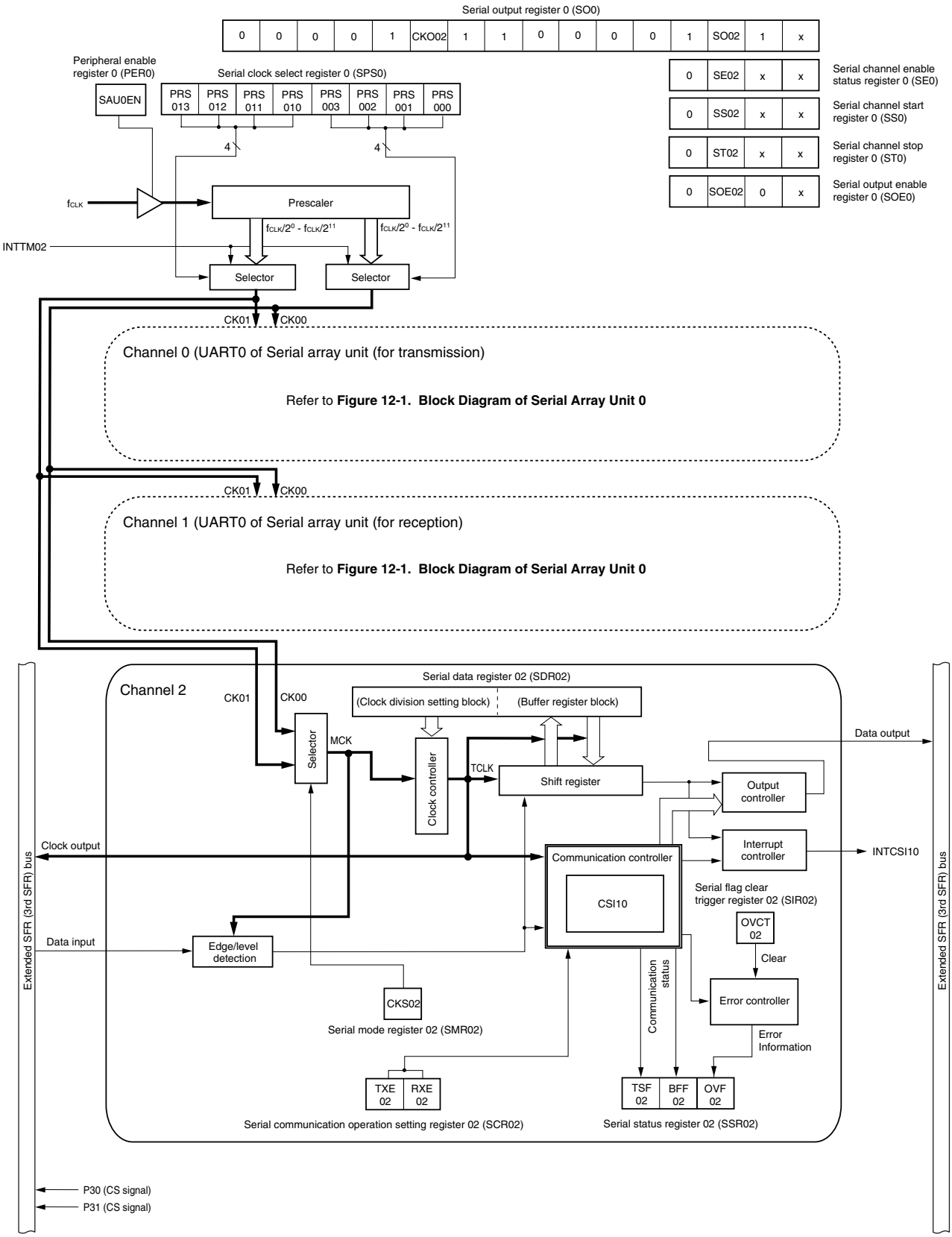
Extended SFR (3rd SFR) interface consists of the following hardware.

Table 14-1. Configuration of Extended SFR (3rd SFR) Interface

Item	Configuration
Shift register	8 bits
Buffer register	Lower 8 bits of serial data register 02 (SDR02)[Note]
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register 0 (SPS0) • Serial channel enable status register 0 (SE0) • Serial channel start register 0 (SS0) • Serial channel stop register 0 (ST0) • Serial output enable register 0 (SOE0) • Serial output register 0 (SO0)
	<Registers of each channel> • Serial data register 02 (SDR02) • Serial mode register 02 (SMR02) • Serial communication operation setting register 02 (SCR02) • Serial status register 02 (SSR02) • Serial flag clear trigger register 02 (SIR02) • Port mode register 1 (PM1) • Port mode register 3 (PM3) • Port register 1 (P1) • Port register 3 (P3)

Note The lower 8 bits of the serial data register 02 (SDR02) can be read or written as the following SFR of SIO10 (CSI10 data register).

Figure 14-1. Block Diagram of the Extended SFR (3rd SFR) Interface



Remark ×: Bit can not be used in extended SFR (3rd SFR) interface.

(1) Shift register

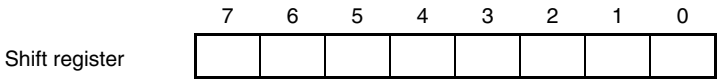
This is an 8-bit register that converts parallel data into serial data or vice versa.

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 8 bits of serial data register 02 (SDR02).



(2) Lower 8 bits of the serial data register 02 (SDR02)

SDR02 is the transmit/receive data register (16 bits) of channel n. Bits 7 to 0 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 (MCK).

When data is received, parallel data converted by the shift register is stored in the lower 8 bits. When data is to be transmitted, set transmit to be transferred to the shift register to the lower 8 bits.

SDR02 can be read or written in 16-bit units.

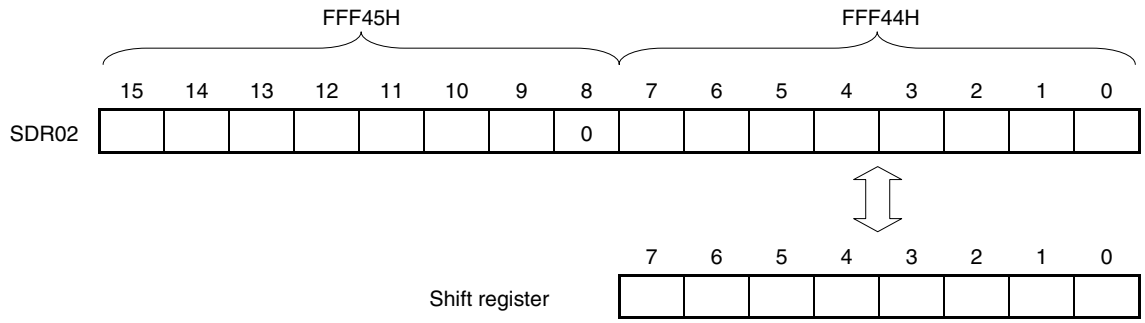
The lower 8 bits of the serial data register 02 (SDR02) can be read or written[Note] as the following SFR of SIO10 (CSI10 data register).

Reset signal generation clears this register to 0000H.

Note Writing in 8-bit units is prohibited when the operation is stopped (when SE02 = 0).

Figure 14-2. Format of Serial Data Register 02 (SDR02)

Address: FFF44H, FFF45H (SDR00), After reset: 0000H R/W



Caution Be sure to clear bit 8 to "0".

14.3 Registers Controlling Extended SFR (3rd SFR) Interface

The extended SFR (3rd SFR) interface is controlled by the following registers.

<Registers of unit setting block (unit 0)>
- Peripheral enable register 0 (PER0)
- Serial clock select register 0 (SPS0)
- Serial channel enable status register 0 (SE0)
- Serial channel start register 0 (SS0)
- Serial channel stop register 0 (ST0)
- Serial output enable register 0 (SOE0)
- Serial output register 0 (SO0)

<Registers of each channel (channel 2)>
- Serial mode register 02 (SMR02)
- Serial communication operation setting register 02 (SCR02)
- Serial data register 02 (SDR02)
- Serial status register 02 (SSR02)
- Serial flag clear trigger register 02 (SIR02)
- Port mode register 1, 3 (PM1, PM3)
- Port register 1, 3 (P1, P3)

(1) Peripheral enable register 0 (PER0)

PER0 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.

In order to use serial array unit 0, be sure to set bit 2 (SAU0EN) of the extended SFR (3rd SFR) interface to 1.

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

Reset signal generation clears this register to 00H.

Figure 14-3. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

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

Cautions 1. When setting the extended SFR (3rd SFR) interface, be sure to set SAU0EN to 1 first. If SAU0EN = 0, writing to control registers of the extended SFR (3rd SFR) interface are ignored, and, even if the registers are read, only the default values are read (except for port mode registers (PM1, PM3), and port registers (P1, P3)).

2. After setting the SAU0EN to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

(2) Serial clock select register 0 (SPS0)

SPS0 is a 16-bit register that is used to select two types of operation clocks (CK00, CK01) that are commonly supplied to each channel. CK01 is selected by bits 7 to 4 of SPS0, and CK00 is selected by bits 3 to 0.

Rewriting SPS0 is prohibited when the register is in operation (when SE02 = 1).

SPS0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SPS0 can be set with an 8-bit memory manipulation instruction with SPS0L.

Reset signal generation clears this register to 0000H.

Figure 14-4. Format of Serial Clock Select Register 0 (SPS0)

Address: F0126H, F0127H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPS0	0	0	0	0	0	0	0	0	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS 0p3	PRS 0p2	PRS 0p1	PRS 0p0	Section of operation clock (CK0p)[Note 1]				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	1	1	INTTM02[Note 2]				
Other than above				Setting prohibited				

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (ST0 = 000FH) the operation of the serial array unit (SAU0). When selecting INTTM02 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAU0 can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting TIS02 of the TIS0 register to 1) and selecting INTTM02 by using the SPS0 register in channels 2 of TAU0. When changing f_{CLK}, however, SAU0 and TAU0 must be stopped as described in Note 1 above.

Cautions 1. Be sure to clear bits 15 to 8 to "0".

2. After setting the SAU0EN to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

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

2. p = 0, 1

(3) Serial channel enable status register 0 (SE0)

SE0 indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written to SS02 of serial channel start register 0 (SS0), the corresponding bit (SE02) of this register is set to 1. When 1 is written to ST02 of serial channel stop register 0 (ST0), the corresponding bit (SE02) is cleared to 0.

Channel 2 that is enabled to operate cannot rewrite by software the value of CKO02 of the serial output register 0 (SO0) to be described below, and a value reflected by a communication operation is output from the serial clock pin.

SE0 can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of SE0 can be set with an 1-bit or 8-bit memory manipulation instruction with SE0L.

Reset signal generation clears this register to 0000H.

Figure 14-5. Format of Serial Channel Enable Status Register 0 (SE0)

Address: F0120H, F0121H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE02	×	×

SE02	Indication of operation enable/stop status of channel 2
0	Operation stops (stops with the values of the control register and shift register, and the statuses of the serial clock I/O pin, serial data output pin, and the OVF error flags retained[Note]).
1	Operation is enabled.

Note Bits 6 and 5 (TSF02, BFF02) of the SSR02 register are cleared.

Caution Be sure to clear bits 15 to 3 of SE0 and bits 15 to 4 of SE1 to "0".

Remark ×: Bit can not be used in extended SFR (3rd SFR) interface.
See **CHAPTER 12 SERIAL ARRAY UNIT.**

(4) Serial channel start register 0 (SS0)

SS0 is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written to SS02, the corresponding bit (SE02) of serial channel enable status register 0 (SE0) is set to 1.

Because SS02 is a trigger bit, it is cleared immediately when SE02 = 1.

SS0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SS0 can be set with an 1-bit or 8-bit memory manipulation instruction with SS0L.

Reset signal generation clears this register to 0000H.

Figure 14-6. Format of Serial Channel Start Register 0 (SS0)

Address: F0122H, F0123H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS02	×	×

SS02	Operation start trigger of channel 2
0	No trigger operation
1	Sets SE02 to 1 and enters the communication wait status (if a communication operation is already under execution, the operation is stopped and the start condition is awaited).

Caution Be sure to clear bits 3 to 15 to “0”.

Remarks 1. When the SS0 register is read, 0000H is always read.

2. ×: Bit can not be used in extended SFR (3rd SFR) interface.
See **CHAPTER 12 SERIAL ARRAY UNIT.**

(5) Serial channel stop register 0 (ST0)

ST0 is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (ST02), the corresponding bit (SE02) of serial channel enable status register 0 (SE0) is cleared to 0. Because ST02 is a trigger bit, it is cleared immediately when SE02 = 0.

ST0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of ST0 can be set with an 1-bit or 8-bit memory manipulation instruction with ST0L.

Reset signal generation clears this register to 0000H.

Figure 14-7. Format of Serial Channel Stop Register 0 (ST0)

Address: F0124H, F0125H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST02	×	×

ST02	Operation stop trigger of channel 2
0	No trigger operation
1	Clears SE02 to 0 and stops the communication operation. (Stops with the values of the control register and shift register, and the statuses of the serial clock I/O pin, serial data output pin, and the OVF error flags retained[Note].)

Note Bits 6 and 5 (TSF02, BFF02) of the SSR02 register are cleared.

Caution Be sure to clear bits 3 to 15 to "0".

Remarks 1. When the ST0 register is read, 0000H is always read.

2. ×: Bit can not be used in extended SFR (3rd SFR) interface.
See **CHAPTER 12 SERIAL ARRAY UNIT.**

(6) Serial output enable register 0 (SOE0)

SOE0 is a register that is used to enable or stop output of the serial communication operation of each channel.

Channel 2 that enables serial output cannot rewrite by software the value of SO02 of the serial output register 0 (SO0) to be described below, and a value reflected by a communication operation is output from the serial data output pin.

SOE0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SOE0 can be set with an 1-bit or 8-bit memory manipulation instruction with SOE0L.

Reset signal generation clears this register to 0000H.

Figure 14-8. Format of Serial Output Enable Register 0 (SOE0)

Address: F012AH, F012BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02	0	×

SOE02	Serial output enable/disable of channel 2
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution Be sure to clear bits 1, 3 to 15 to "0".

Remark ×: Bit can not be used in extended SFR (3rd SFR) interface.
See **CHAPTER 12 SERIAL ARRAY UNIT.**

(7) Serial output register 0 (SO0)

SO0 is a buffer register for serial output of each channel.

The value of bit 2 of this register is output from the serial data output pin of channel 2.

The value of bit (2 + 8) of this register is output from the serial clock output pin of channel 2.

SO02 of this register can be rewritten by software only when serial output is disabled (SOE02 = 0). When serial output is enabled (SOE02 = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

CKO02 of this register can be rewritten by software only when the channel operation is stopped (SE02 = 0). While channel operation is enabled (SE02 = 1), rewriting by software is ignored, and the value of CKOmn can be changed only by a serial communication operation.

SO0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0F0FH.

Figure 14-9. Format of Serial Output Register 0 (SO0)

Address: F0128H, F0129H 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	CKO02	1	1	0	0	0	0	1	SO02	1	×

CKO02	Serial clock output of channel 2
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

SO02	Serial data output of channel 2
0	Serial data output value is “0”.
1	Serial data output value is “1”.

Caution Be sure to set bits 1, 3, 8, 9, and 11 to “1”.
And be sure to clear bits 4 to 7, 12 to 15 to “0”.

Remark ×: Bit can not be used in extended SFR (3rd SFR) interface.
See **CHAPTER 12 SERIAL ARRAY UNIT.**

(8) Serial mode register 02 (SMR02)

SMR02 is a register that sets an operation mode of channel n. It is also used to select an operation clock (MCK) and an interrupt source.

Rewriting SMR02 is prohibited when the register is in operation (when SE02 = 1). However, the MD020 bit can be rewritten during operation.

SMR02 can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets this register to 0020H.

Figure 14-10. Format of Serial Mode Register 02 (SMR02)

Address: F0114H, F0115H After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR02	CKS02	0	0	0	0	0	0	0	0	0	1	0	0	0	0	MD020

CKS02	Selection of operation clock (MCK) of channel 2
0	Prescaler output clock CK00 set by PRS register
1	Prescaler output clock CK01 set by PRS register
Operation clock MCK is used by the edge detector. In addition, depending on the higher 7 bits of the SDR02 register, a transfer clock (TCLK) is generated.	

MD020	Selection of interrupt source of channel 2
0	Transfer end interrupt
1	Buffer empty interrupt
For successive transmission, the next transmit data is written by setting MD020 to 1 when SDR02 data has run out.	

Caution Be sure to clear bits 1 to 4, 6 to 14 to "0". Be sure to set bit 5 to "1".

(9) Serial communication operation setting register 02 (SCR02)

SCR02 is a communication operation setting register of channel 2. It is used to set a data transmission/reception mode.

Rewriting SCR02 is prohibited when the register is in operation (when SE02 = 1).

SCR02 can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets this register to 0087H.

Figure 14-11. Format of Serial Communication Operation Setting Register 02 (SCR02)

Address: F011CH, F011DH After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR02	TXE02	RXE02	0	0	0	0	0	0	0	0	0	0	0	1	1	1

TXE02	RXE02	Setting of operation mode of channel 2
0	0	Does not start communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

Cautions 1. Be sure to clear bits 3 to 13 to "0". Be sure to set bit 0 to 2 to "1".

2. Be sure to clear bit 7 (DIR02) to "0" after reset release.

(10) Higher 7 bits of the serial data register 02 (SDR02)

SDR02 is the transmit/receive data register (16 bits) of channel 2. Bits 7 to 0 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 (MCK).

The clock set by dividing the operating clock by the higher 7 bits of SDR02 is used as the transfer clock.

For the function of the lower 8 bits of SDR02, see **14.2 Configuration of Extended SFR (3rd SFR) Interface**.

SDR02 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 (SE02 = 0). During operation (SE02 = 1), a value is written only to the lower 8 bits of SDR02. When SDR02 is read during operation, 0 is always read.

Reset signal generation clears this register to 0000H.

Figure 14-12. Format of Serial Data Register 02 (SDR02)

Address: FFF44H, FFF45H After reset: 0000H R/W

	FFF45H								FFF44H							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR02								0								

SDR02[15:9]							Transfer clock setting by dividing the operating clock (MCK)
0	0	0	0	0	0	0	MCK/2
0	0	0	0	0	0	1	MCK/4
0	0	0	0	0	1	0	MCK/6
0	0	0	0	0	1	1	MCK/8
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	MCK/254
1	1	1	1	1	1	1	MCK/256

Cautions 1. Be sure to clear bit 8 to "0".

<R> **2. Transfer clock frequency for the extended SFR (3rd SFR) interface must be 1/4 or less of the clock frequency supplied to the interface, and must satisfy the following.**

- **When $V_{DD} = LV_{DD} \geq 2.7$ V : Set transfer clock frequency to 1.25 MHz or less**
- **When $V_{DD} = LV_{DD} < 2.7$ V : Set transfer clock frequency to 555 kHz or less**

Remark The lower 8 bits of the serial data register 02 (SDR02) can be read or written as the following SFR of SIO10 (CSI10 data register).

(11) Serial status register 02 (SSR02)

SSR02 is a register that indicates the communication status and error occurrence status of channel 2. The errors indicated by this register is overrun error.

SSR02 can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of SSR02 can be set with an 8-bit memory manipulation instruction with SSR02L.

Reset signal generation clears this register to 0000H.

Figure 14-13. Format of Serial Status Register 02 (SSR02)

Address: F0104H, F0105H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSR02	0	0	0	0	0	0	0	0	0	TSF02	BFF02	0	0	0	0	OVF02

TSF02	Communication status indication flag of channel 2
0	Communication is not under execution.
1	Communication is under execution.
Because this flag is an updating flag, it is automatically cleared when the communication operation is completed. This flag is cleared also when the ST02/SS02 bit is set to 1.	

BFF02	Buffer register status indication flag of channel 2
0	Valid data is not stored in the SDR02 register.
1	Valid data is stored in the SDR02 register.
This is an updating flag. It is automatically cleared when transfer from the SDR02 register to the shift register is completed. During reception, it is automatically cleared when data has been read from the SDR02 register. This flag is cleared also when the ST02/SS02 bit is set to 1. This flag is automatically set if transmit data is written to the SDR02 register when the TXE02 bit of the SCR02 register = 1 (transmission or reception mode in each communication mode). It is automatically set if receive data is stored in the SDR02 register when the RXE02 bit of the SCR02 register = 1 (transmission or reception mode in each communication mode). It is also set in case of a reception error. If data is written to the SDR02 register when BFF02 = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVF02 = 1) is detected.	

OVF02	Overrun error detection flag of channel 2
0	No error occurs.
1	An overrun error occurs. <Causes of overrun error> • Receive data stored in the SDR02 register is not read and transmit data is written or the next receive data is written. • Transmit data is not ready for slave transmission or reception in the CSI mode.
This is a cumulative flag and is not cleared until 1 is written to the OVCT02 bit of the SIR02 register.	

(12) Serial flag clear trigger register 02 (SIR02)

SIR02 is a trigger register that is used to clear each error flag of channel 2.

When OVCT02 of this register is set to 1, the corresponding bit (OVF02) of serial status register 02 (SSR02) is cleared to 0. Because SIR02 is a trigger register, it is cleared immediately when the corresponding bit of SSR02 is cleared.

SIR02 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SIR02 can be set with an 8-bit memory manipulation instruction with SIR02L.

Reset signal generation clears this register to 0000H.

Figure 14-14. Format of Serial Flag Clear Trigger Register 02 (SIR02)

Address: F010CH, F010DH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIR02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVCT02

OVCT02	Clear trigger of overrun error flag of channel 2
0	No trigger operation
1	Clears the OVF02 bit of the SSR02 register to 0.

Caution Be sure to clear bits 1 to 15 to "0".

Remark When the SIR02 register is read, 0000H is always read.

(13) Port mode registers 1 (PM1)

This register is used to set input/output of ports 1 in 1-bit units.

Port mode register PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 14-15. Format of Port Mode Registers 1 (PM1)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	PM1n pin I/O mode selection (n = 0 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

<R> **Cautions 1. Be sure to clear PM13 and PM15 to "0", and to set PM14 to "1", after reset release.**

2. Be sure to set bit 7 to "1".

(14) Port mode registers 3 (PM3)

This register is used to set input/output of ports 3 in 1-bit units.

After reset is released, be sure to clear PM30, PM31, and PM32 to "0".

Port mode register PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 14-16. Format of Port Mode Registers 3 (PM3)

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30

PM3n	PM3n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Cautions 1. Be sure to clear PM30, PM31, and PM32 to "0" after reset release.

2. Be sure to set bits 5 to 7 to "1".

(15) Port register 1 (P1)

This register is used to write the data that is output from the chip when data is output from a port.

After reset is released, be sure to set P15 to 1.

Port register P1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 14-17. Format of Port Register 1 (P1)

Address: FF01H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
P1	0	P16	P15	P14	P13	P12	P11	P10

P1n	n = 0 to 6	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Caution Be sure to set P15 to "1" after reset release.

(16) Port register 3 (P3)

P30 and P31 are chip select signals that select the extended SFR (3rd SFR) space. The range of addresses accessed in the extended SFR (3rd SFR) is specified by the combination of P30 and P31.

Figure 14-18. Format of Port Register 3 (P3)

Address: FF03H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
P3	0	0	0	P34	P33	P32	P31	P30

P31	P30	Extended SFR (3rd SFR) address
0	0	Access prohibited
0	1	80H to FFH
1	0	100H to 17FH
1	1	180H to 1FFH

14.4 Operation of Extended SFR (3rd SFR) Interface

Extended SFR (3rd SFR) interface has the following two modes.

- Operation stop mode
- Serial I/O mode

14.4.1 Operation stop mode

Extended SFR (3rd SFR) is not accessed in this mode. Therefore, the power consumption can be reduced.

(1) Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

PER0 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.

Set bit 2 (SAU0EN) to 0 so that the extended SFR (3rd SFR) interface stops by stopping serial array unit 0.

Figure 14-19. Peripheral Enable Register 0 (PER0) Setting When Stopping Operation by Units

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAU0 to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	RTCEN		ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
	×	0	×	×	×	0/1	×	×

Control of SAU0 input clock
0: Stops supply of input clock
1: Supplies input clock

Cautions 1. If SAU0EN = 0, writing to a control register of the extended SFR (3rd SFR) interface is ignored, and, even if the register is read, only the default value is read (except for port mode register 3 (PM3), and port register 3 (P3)).

2. Be careful when setting SAU0EN = 0 because serial communication (UART0) of other channels of serial array unit 0 also stops.

Remark ☐ : Setting disabled (set to the initial value)

×: Bits not used with the extended SFR (3rd SFR) interface (depending on the settings of other peripheral functions)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Stopping the operation by channels

The extended SFR (3rd SFR) can stop channel 2 of serial array unit 0. The stopping of the operation by channels is set using each of the following registers.

Figure 14-20. Each Register Setting When Stopping Operation by Channels

(a) Serial Channel Enable Status Register 0 (SE0) …

This register is used to indicate whether data transmission/reception operation of channel 2 is enabled or stopped.

SE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	SE02	SE01	SE00
													0/1	×	×

0: Operation stops

- The SE0 register is a read-only status register, whose operation is stopped by using the ST0 register.

With a channel whose operation is stopped, the value of CKO02 of the SO0 register can be set by software.

(b) Serial channel stop register 0 (ST0) …

This register is a trigger register that is used to enable stopping communication/count of channel 2.

ST0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	ST02	ST01	ST00
													0/1	×	×

1: Clears SE02 to 0 and stops the communication operation

* Because ST02 is a trigger bit, ST02 is cleared immediately when SE02 = 0.

(c) Serial output enable register 0 (SOE0) …

This register is used to enable or stop output of the serial communication operation of channel 2.

SOE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02		SOE00
													0/1	0	×

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SO02 value of the SO0 register can be set by software.

(d) Serial output register 0 (SO0) …

This register is a buffer register for serial output of channel 2.

SO0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	CKO02	1	1	0	0	0	0	1	SO02		SO00
					0/1								0/1	1	×

1: Serial clock output value is “1” 1: Serial data output value is “1”

Remark : Setting disabled (set to the initial value),

×: Bit can not be used in extended SFR (3rd SFR) interface.

See **CHAPTER 12 SERIAL ARRAY UNIT.**

0/1: Set to 0 or 1 depending on the usage of the user

14.4.2 Serial I/O mode

The 1-byte data transmission/reception is executed.

(1) Overview of functions

Serial I/O	CSI10
Target channel	Channel 2 of SAU0
Internal signal	$\overline{\text{SCK10}}$, SI10, SO10
Interrupt	INTCS10
	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 (OVF02) only
Transfer data length	8 bits
Transfer rate	Max. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz]Note f_{CLK}: System clock frequency
Data phase	By DAP02 = 0, Data I/O starts from the start of the operation of the serial clock.
Clock phase	By CKP02 = 0, it is forward.
Data direction	MSB first

<R> **Note** Transfer clock frequency for the extended SFR (3rd SFR) interface must be 1/4 or less of the clock frequency supplied to the interface, and must satisfy the following.

- When $V_{DD} = LV_{DD} \geq 2.7$ V : Set transfer clock frequency to 1.25 MHz or less
- When $V_{DD} = LV_{DD} < 2.7$ V : Set transfer clock frequency to 555 kHz or less

(2) Register setting

Figure 14-21. Example of Contents of Registers on Serial I/O Mode

(a) Serial output register 0 (SO0) … Sets only the bits of the channel 2.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 0/1	1	1	0	0	0	0	1	SO02 0/1	1	SO00 ×

Communication starts when this bit is 1 because the data phase is forward (CKP02 = 0).

(b) Serial output enable register 0 (SOE0) … Sets only the bits of the channel 2 to 1.

	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	SOE02 0/1	0	SOE00 ×

(c) Serial channel start register 0 (SS0) … Sets only the bits of the channel 2 to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS02 0/1	SS01 ×	SS00 ×

(d) Serial mode register 02 (SMR02)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR02	CKS02 0/1	CCS02 0	0	0	0	0	0	STS02 0	0	SIS02 0	1	0	0	MD022 0	MD021 0	MD020 0/1

Operation mode of channel 2
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 02 (SCR02)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR02	TXE02 0/1	RXE02 0/1	DAP02 0	CKP02 0	0	EOC02 0	PTC021 0	PTC020 0	DIR02 0	0	SLC021 0	SLC020 0	0	DLS022 1	DLS021 1	DLS020 1

(f) Serial data register 02 (SDR02) (lower 8 bits: SIO10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR02	Baud rate setting							0	Transmit data setting/reception data register							

SIO10

Remark ☐ : Setting is fixed in serial I/O mode of the extended SFR (3rd SFR) interface,
▒ : Setting disabled (set to the initial value)
×: Bit can not be used in extended SFR (3rd SFR) interface.
See **CHAPTER 12 SERIAL ARRAY UNIT.**
0/1: Set to 0 or 1 depending on the usage of the user

14.5 Reading from and Writing to Extended SFR (3rd SFR)

The extended SFR (3rd SFR) space contains 384 bytes and is divided into 128-byte banks. A combination of bank selection and serial communication is used to access the extended SFR (3rd SFR) for reading or writing.

(1) Specifying an extended SFR (3rd SFR) bank

The extended SFR (3rd SFR) space consists of three 128-byte banks. Banks are selected by using the bank selection signals (P30 and P31). The bank that corresponds to each chip select signal is as follows.

Table 14-2. Bank That Corresponds to Each Chip Select Signal

P31	P30	Extended SFR (3rd SFR) address
0	0	Access prohibited
0	1	80H to FFH
1	0	100H to 17FH
1	1	180H to 1FFH

Caution Changing the bank selection signal during serial communication (when TSF02 of serial status register 02 (SSR02) is 1) is prohibited.

(2) Commands and data

Commands are used to specify extended SFR (3rd SFR)s to access.

Commands are used to specify whether to access the extended SFR (3rd SFR)s for reading or writing and to specify a bank-internal address. The type of access is specified for the most significant bit (bit 7), and the bank-internal address is specified for bits 6 to 0.

The following shows the format of access commands and data.

Figure 14-22. Access Command Format

	7	6	5	4	3	2	1	0
Command	R/W	A6	A5	A4	A3	A2	A1	A0

R/W	Specify register reading or writing
0	Reading (This setting specifies reading from the extended SFR (3rd SFR)).
1	Writing (This setting specifies writing to the extended SFR (3rd SFR)).

A6 to A0	Specify the lower 7 bits of an extended SFR (3rd SFR) address.

Figure 14-23. Data Format

	7	6	5	4	3	2	1	0
Data	D7	D6	D5	D4	D3	D2	D1	D0

D7 to D0	8-bit data to be transmitted or received.

(3) Reading extended SFR (3rd SFR)s

The data in extended SFR (3rd SFR)s is accessed for reading by performing the following procedure:

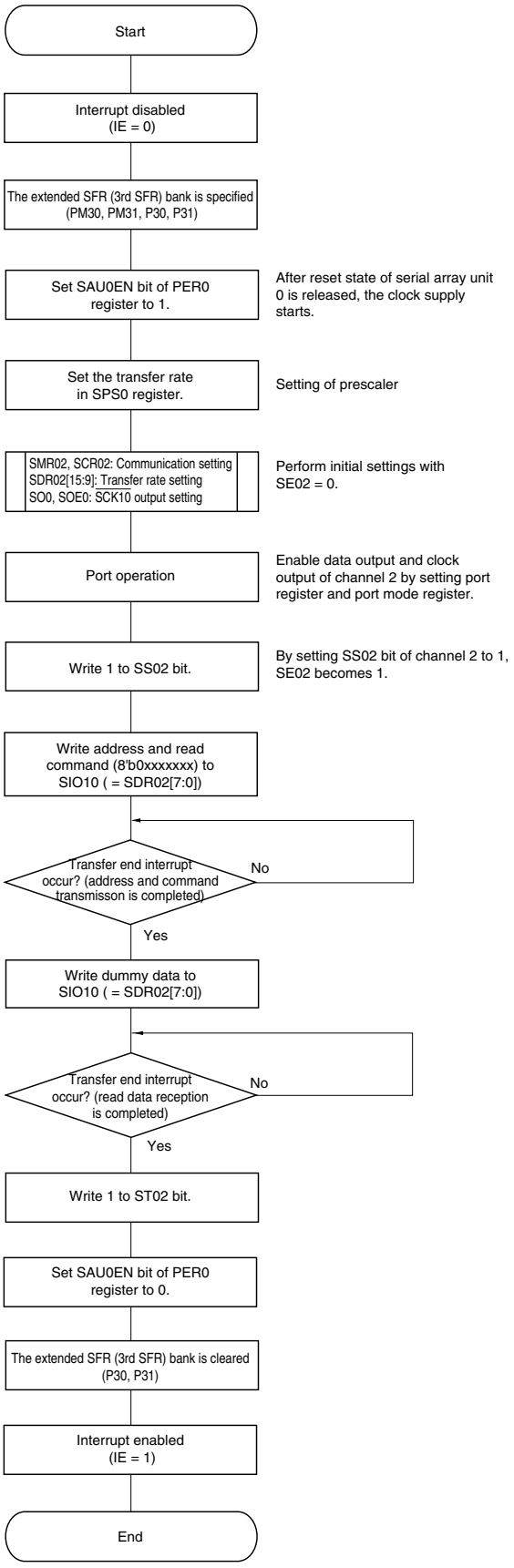
<1> Selecting an extended SFR (3rd SFR) bank (P30, P31)
<2> Transmitting an access command
<3> Receiving read data
<4> Clearing the extended SFR (3rd SFR) bank selection (P30, P31)

Cautions 1. Be sure to clear the SFR banks if accessing non-sequential addresses, even if the addresses are in the same bank.
2. Only access the extended SFR (3rd SFR) register while interrupts are disabled (IE = 0).

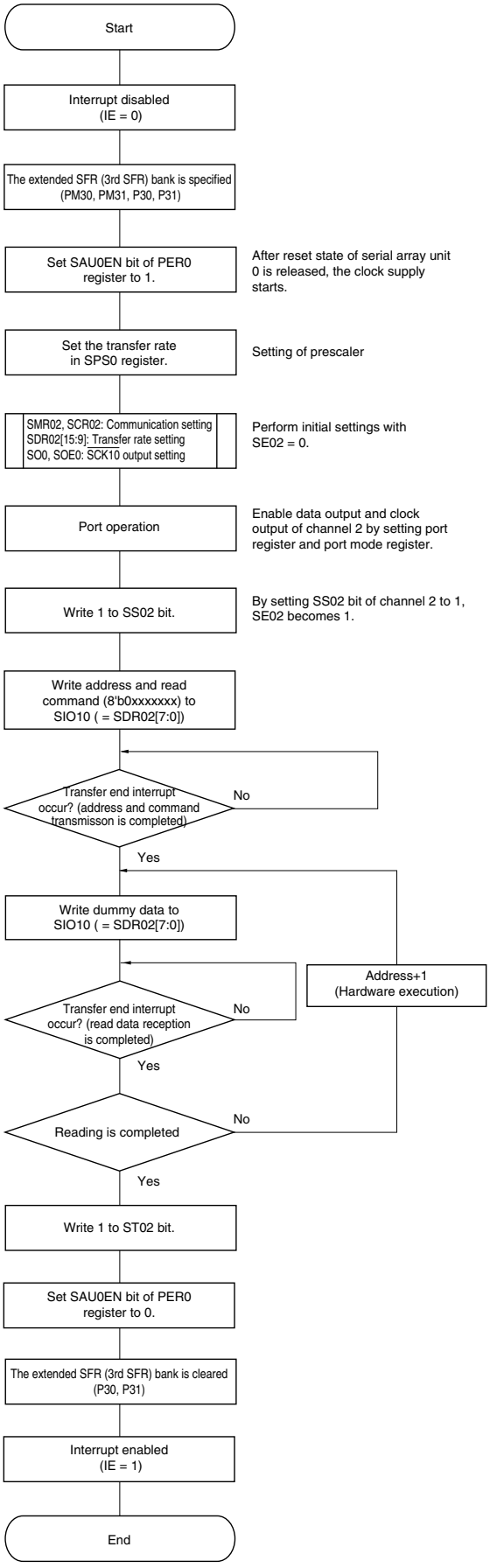
Note that, if reading data at sequential addresses, the second and subsequent command transmissions can be omitted. After the data reading in step <3> finishes, the extended SFR (3rd SFR) interface increments the address and sequentially reads the data. Because the processing to issue commands can be omitted, data can be read at high speed. The relationship between the register settings and pins is shown below.

Figure 14-24. Accessing Data in Extended SFR (3rd SFR)s for Reading

(a) If reading non-sequential addresses



(b) If reading sequential addresses



(4) Writing access

The data in extended SFR (3rd SFR) is accessed for writing by performing the following procedure:

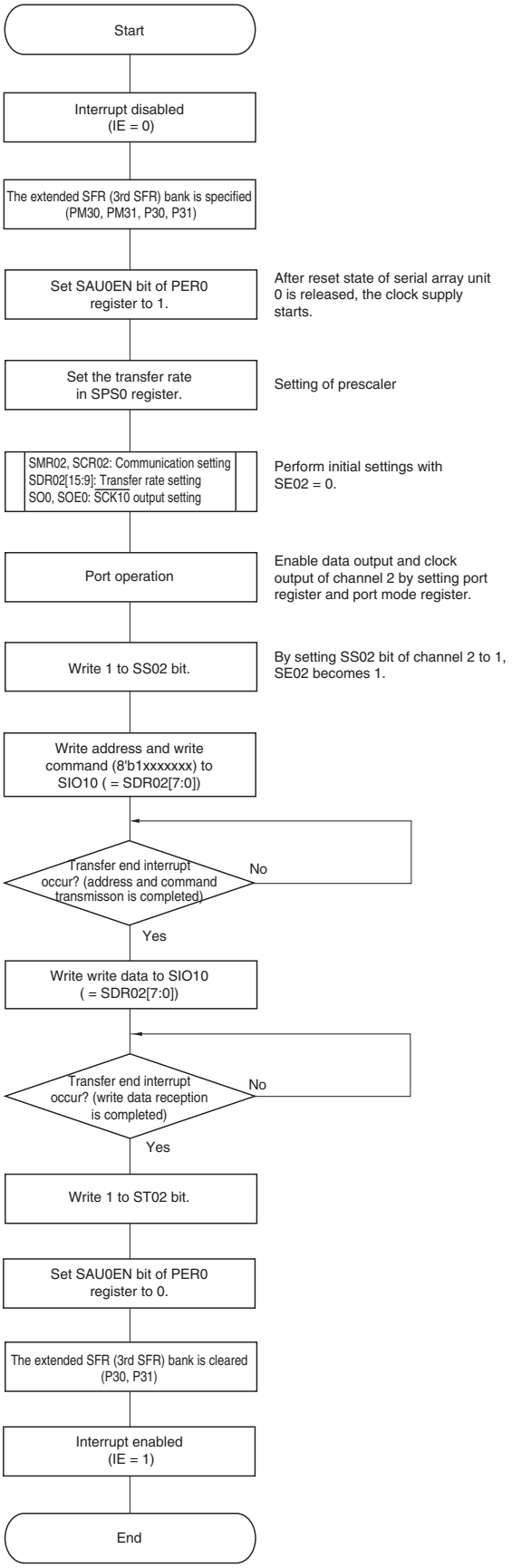
<1> Selecting an extended SFR (3rd SFR) bank (P30, P31)
<2> Transmitting an access command
<3> Transmitting write data
<4> Clearing the extended SFR (3rd SFR) bank selection (P30, P31)

Cautions 1. Be sure to clear the SFR banks if accessing non-sequential addresses, even if the addresses are in the same bank.
2. Only access the extended SFR (3rd SFR) register while interrupts are disabled (IE = 0).
3. If an internal reset is triggered while using the extended SFR (3rd SFR) interface to transmit data, the data might not be correctly transmitted to the extended SFR (3rd SFR).

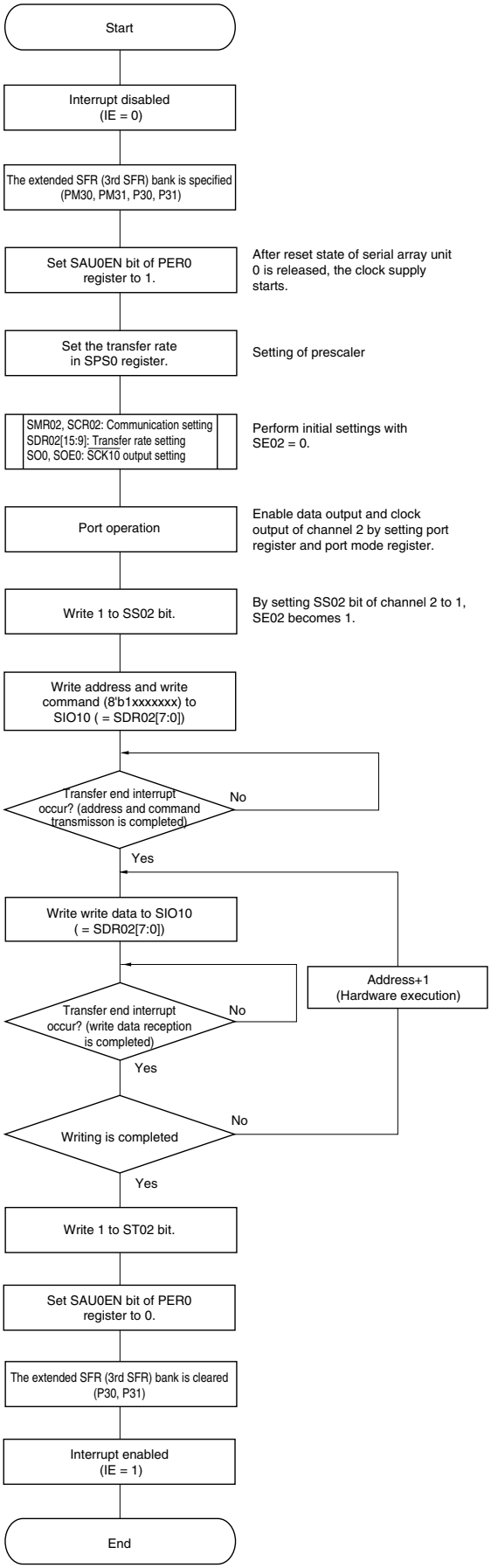
Note that, if writing data at sequential addresses, the second and subsequent command transmissions can be omitted. Repeating the write data transmission in step <3> makes it possible to continuously write data. Because the processing to issue commands can be omitted, data can be written at high speed.

Figure 14-25. Accessing Data in Extended SFR (3rd SFR)s for Writing

(a) If writing non-sequential addresses



(b) If writing sequential addresses



CHAPTER 15 LCD CONTROLLER/DRIVER

15.1 Functions of LCD Controller/Driver

The functions of the LCD controller/driver in the 78K0R/Lx3-M microcontrollers are as follows.

(1) The LCD driver voltage generator can switch internal voltage boosting method, capacitor split method, and external resistance division method.
(2) Automatic output of segment and common signals based on automatic display data memory read
(3) Five different display modes:
- Static
- 1/2 duty (1/2 bias)
- 1/3 duty (1/2 bias)
- 1/3 duty (1/3 bias)
- 1/4 duty (1/3 bias)

(4) Five different frame frequencies, selectable in each display mode
(5) The reference voltage to be generated when operating the voltage boost circuit can be selected from 20 stages (contrast adjustment).
(6) The data display of the LCD display data memory can be selected from three types.
- Displaying an A-pattern area (lower four bits)
- Displaying a B-pattern area (higher four bits)
- Alternately displaying A-pattern and B-pattern areas (blinking display corresponding to the constant-period interrupt (INTRTC) timing of the real-time counter 2 (RTC2))

(7) Segment signal outputs: 40 (SEG0 to SEG39), common signal outputs: 4 (COM0 to COM3)

Table 15-1 lists the maximum number of pixels that can be displayed in each display mode.

Table 15-1. Maximum Number of Pixels

LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Common Signals Used	Number of Segments	Maximum Number of Pixels
External resistance division	–	Static	COM0 (COM1 to COM3)	40	40 (40 segment signals, 1 common signal)Note 1
	1/2	2	COM0, COM1		80 (40 segment signals, 2 common signals)Note 2
		3	COM0 to COM2		120 (40 segment signals, 3 common signals)Note 3
	1/3	3	COM0 to COM2		
		4	COM0 to COM3		160 (40 segment signals, 4 common signals)Note 4
Internal voltage boosting	1/3	3	COM0 to COM2		120 (40 segment signals, 3 common signals)Note3
		4	COM0 to COM3		160 (40 segment signals, 4 common signals)Note 4
capacitor split	1/3	3	COM0 to COM2		120 (40 segment signals, 3 common signals)Note 3
		4	COM0 to COM3		160 (40 segment signals, 4 common signals)Note 4

<R> **Notes 1.** 5-digit display on a 8. configuration LCD panel (using 8 segments × 1 common)
2. 10-digit display on a 8. configuration LCD panel (using 4 segments × 2 commons)
3. 15-digit display on a 8. configuration LCD panel (using 3 segments × 3 commons)
4. 20-digit display on a 8. configuration LCD panel (using 2 segments × 4 commons)

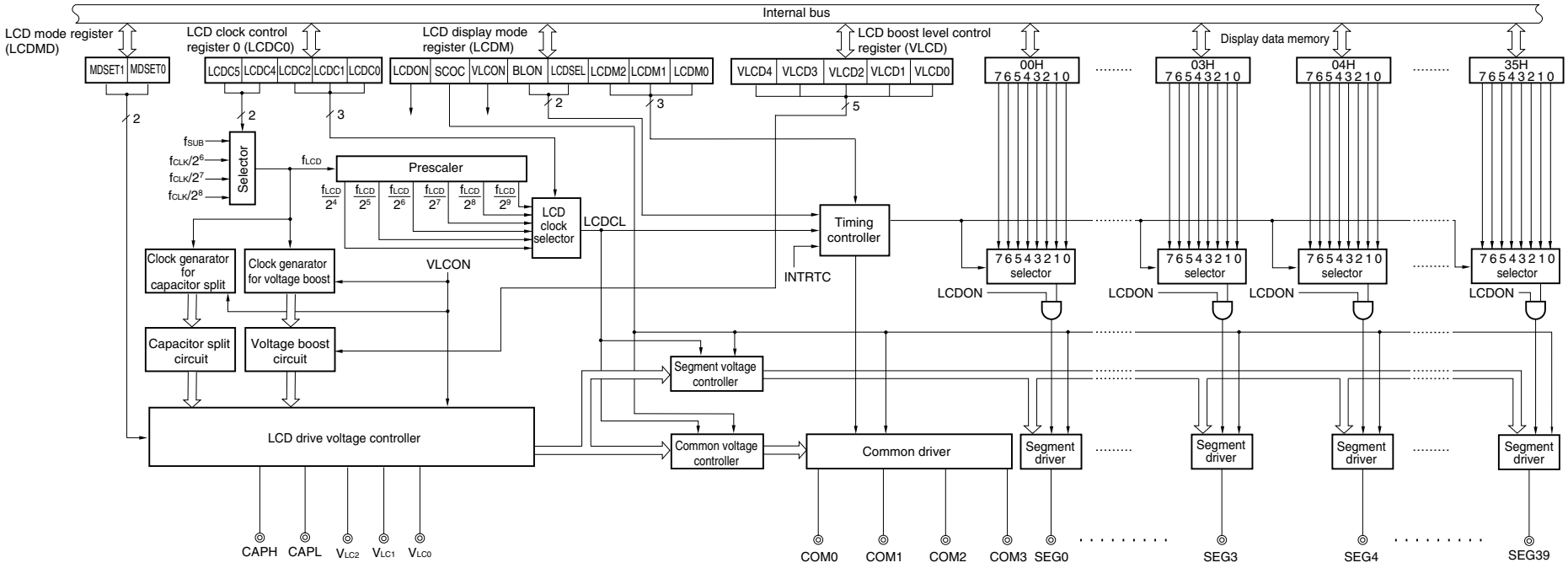
15.2 Configuration of LCD Controller/Driver

The LCD controller/driver consists of the following hardware.

Table 15-2. Configuration of LCD Controller/Driver

Item	Configuration
Display outputs	Segment signal outputs: 40 (SEG0 to SEG39), common signal outputs: 4 (COM0 to COM3)
Control registers	LCD mode register (LCDMD) LCD display mode register (LCDM) LCD clock control register 0 (LCDC0) LCD boost level control register (VLCD) Port function register (PFALL) Segment enable register (SEGEN) Input switch control register (ISC)

Figure 15-1. Block Diagram of LCD Controller/Driver



Remark Segment signal outputs: 40 (SEG0 to SEG39), common signal outputs: 4 (COM0 to COM3)

15.3 Registers Controlling LCD Controller/Driver

The following seven registers are used to control the LCD controller/driver.

- LCD mode register (LCDMD)
- LCD display mode register (LCDM)
- LCD clock control register 0 (LCDC0)
- LCD boost level control register (VLCD)
- Port function register (PFALL)
- Segment enable register (SEGEN)
- Input switch control register (ISC)

(1) LCD mode register (LCDMD)

LCDMD sets the LCD drive voltage generator.

LCDMD is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDMD to 00H.

Figure 15-2. Format of LCD Mode Register (LCDMD)

Address: FFF40H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDMD	0	0	MDSET1	MDSET0	0	0	0	0

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

Caution Bits 0 to 3, 6 and 7 must be set to 0.

(2) LCD display mode register (LCDM)

LCDM is a register that enables/disables display operation, enables/disables voltage boost circuit or capacitor split circuit operation, and sets the display data area and the display mode.

LCDM is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDM to 00H.

Figure 15-3. Format of LCD Display Mode Register (LCDM)

Address: FFF41H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
LCDM	LCDON	SCOC	VLCON	BLON	LCDSEL	LCDM2	LCDM1	LCDM0

LCDON	SCOC	LCD display enable/disable
0	0	Output ground level to segment/common pin
0	1	Display off (all segment outputs are deselected.)
1	0	Output ground level to segment/common pin
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
1	Enables voltage boost circuit or capacitor split circuit operation

BLON	LCDSEL	Display data area control
0	0	Displaying an A-pattern area data (lower four bits of LCD display data memory)
0	1	Displaying a B-pattern area data (higher four bits of LCD display data memory)
1	0	Alternately displaying A-pattern and B-pattern area data (blinking display corresponding to
1	1	the constant-period interrupt (INTRTC) timing of the real-time counter 2 (RTC2))

LCDM2	LCDM1	LCDM0	LCD controller/driver display mode selection					
			External resistance division method		Internal voltage boosting method		Capacitor split method	
			Number of time slices	Bias mode	Number of time slices	Bias mode	Number of time slices	Bias mode
0	0	0	4	1/3	4	1/3	4	1/3
0	0	1	3	1/3	3	1/3	3	1/3
0	1	0	2	1/2	4	1/3	4	1/3
0	1	1	3	1/2	4	1/3	4	1/3
1	0	0	Static		Setting prohibited			
Other than above			Setting prohibited					

Cautions 1. When LCD display is not performed or necessary, set SCOC and VLCON to 0, in order to reduce power consumption.

2. When the external resistance division method has been set (MDSET1 = MDSET0 = 0), do not set VLCON to 1.

3. To use the internal voltage boosting method, specify the reference voltage by using the VLCD register (or perform a reset to use the default value of the reference voltage), wait for the reference voltage setup time (2 ms (min.)), and then set VLCON to 1.

Caution 4. To manipulate VLCON when using the internal voltage boosting method or capacitor split method, follow the procedure below.

A. To stop the operation of the voltage boosting/capacitor split circuit after switching display status from on to off:

1) Set to display off status by setting LCDON = 0.

2) Disable outputs of all the segment buffers and common buffers by setting SCOC = 0.

3) Stop the operation of the voltage boosting/capacitor split circuit by setting VLCON = 0.

B. To stop the operation of the voltage boosting/capacitor split circuit during display on status:

Setting prohibited. Be sure to stop the operation of the voltage boosting/capacitor split circuit after setting display off.

C. To set display on from stop status of the voltage boosting/capacitor split circuit:

1) Start the operation of the voltage boosting/capacitor split circuit by setting VLCON = 1, then wait for the voltage boosting/capacitor split wait time (see CHAPTER 32 ELECTRICAL SPECIFICATIONS).

2) Set all the segment buffers and common buffers to non-display output status by setting SCOC = 1.

3) Set display on by setting LCDON = 1.

(3) LCD clock control register (LCDC0)

LCDC0 specifies the LCD source clock and LCD clock.

The frame frequency is determined according to the LCD clock and the number of time slices.

LCDC0 is set using an 8-bit memory manipulation instruction.

Reset signal generation sets LCDC0 to 00H.

Figure 15-4. Format of LCD Clock Control Register (LCDC0)

Address: FFF42H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDC0	0	0	LCDC05	LCDC04	0	LCDC02	LCDC01	LCDC00

LCDC05	LCDC04	LCD source clock (f_{LCD}) selection
0	0	f_{SUB}
0	1	$f_{CLK}/2^6$
1	0	$f_{CLK}/2^7$
1	1	$f_{CLK}/2^8$

LCDC02	LCDC01	LCDC00	LCD clock (LCDCL) selection
0	0	0	$f_{LCD}/2^4$
0	0	1	$f_{LCD}/2^5$
0	1	0	$f_{LCD}/2^6$
0	1	1	$f_{LCD}/2^7$
1	0	0	$f_{LCD}/2^8$
1	0	1	$f_{LCD}/2^9$
Other than above			Setting prohibited

Cautions 1. Bits 3, 6, and 7 must be set to 0.

2. Set the LCD clock (LCDCL) to no more than 512 Hz when the internal voltage boost method has been set.

Remark f_{CLK}: CPU/Peripheral hardware clock frequency

f_{SUB}: Subsystem clock frequency

(4) LCD boost level control register (VLCD)

This register is used to select the reference voltage that is to be generated when operating the voltage boost circuit (contrast adjustment). The reference voltage can be selected from 20 stages.

VLCD is set using an 8-bit memory manipulation instruction.

Reset signal generation sets VLCD to 0FH.

Figure 15-5. Format of LCD Boost Level Control Register (VLCD)

Address: FFF43H After reset: 0FH 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)	V_{LC0} voltage 1/3 bias
0	0	0	0	0	1.75 V	5.25 V
0	0	0	0	1	1.70 V	5.10 V
0	0	0	1	0	1.65 V	4.95 V
0	0	0	1	1	1.60 V	4.80 V
0	0	1	0	0	1.55 V	4.65 V
0	0	1	0	1	1.50 V	4.50 V
0	0	1	1	0	1.45 V	4.35 V
0	0	1	1	1	1.40 V	4.20 V
0	1	0	0	0	1.35 V	4.05 V
0	1	0	0	1	1.30 V	3.90 V
0	1	0	1	0	1.25 V	3.75 V
0	1	0	1	1	1.20 V	3.60 V
0	1	1	0	0	1.15 V	3.45 V
0	1	1	0	1	1.10 V	3.30 V
0	1	1	1	0	1.05 V	3.15 V
0	1	1	1	1	1.00 V (default)	3.00 V
1	0	0	0	0	0.95 V	2.85 V
1	0	0	0	1	0.90 V	2.70 V
1	0	0	1	0	0.85 V	2.55 V
1	0	0	1	1	0.80 V	2.40 V
Other than above					Setting prohibited	

Cautions 1. The VLCD setting is valid only when the voltage boost circuit is operating.

2. Bits 5 to 7 must be set to 0.

3. Be sure to change the VLCD value after having stopped the operation of the voltage boost circuit (VLCON = 0).

4. These values above may change after device evaluation.

5. To use the internal voltage boosting method, specify the reference voltage by using the VLCD register (or perform a reset to use the default value of the reference voltage), wait for the reference voltage setup time (2 ms (min.)), and then set VLCON to 1.

(5) Port function register (PFALL)

This register sets whether to use pins P50 to P57, P90 to P97, P100, and P140 to P147 as port pins (other than segment output pins) or segment output pins.

PFALL is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PFALL to 00H.

Figure 15-6. Format of Port Function Register (PFALL)

Address: F0080H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PFALL	0	PF14H	PF14L	PF10	PF9H	PF9L	PF5H	PF5L

PF14H	Port/segment outputs specification of the P144 to P147 pins
0	Used the P144 to P147 pins as port (other than segment output)
1	Used the P144 to P147 pins as segment output

PF14L	Port/segment outputs specification of the P140 to P143 pins
0	Used the P140 to P143 pins as port (other than segment output)
1	Used the P140 to P143 pins as segment output

PF10	Port/segment outputs specification of the P100 pin
0	Used the P100 pin as port (other than segment output)
1	Used the P100 pin as segment output

PF9H	Port/segment outputs specification of the P94 to P97 pins
0	Used the P94 to P97 pins as port (other than segment output)
1	Used the P94 to P97 pins as segment output

PF9L	Port/segment outputs specification of P90 to P93 pins
0	Used the P90 to P93 pins as port (other than segment output)
1	Used the P90 to P93 pins as segment output

PF5H	Port/segment outputs specification of the P54 to P57 pins
0	Used the P54 to P57 pins as port (other than segment output)
1	Used the P54 to P57 pins as segment output

PF5L	Port/segment outputs specification of P50 to P53 pins
0	Used the P50 to P53 pins as port (other than segment output)
1	Used the P50 to P53 pins as segment output

Caution Be sure to clear bit 7 “0”.

(6) Segment enable register (SEGEN)

SEGEN is a register that is used to enable or disable segment output to segment output only pins (SEG8 to SEG14).

SEGEN is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets SEGEN to 00H.

Figure 15-7. Format of Segment Enable Register (SEGEN)

Address: F0081H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
SEGEN	0	0	0	0	0	0	SEGEN1	SEGEN0

SEGENn	Output enable/disable to segment output only pins (n = 0, 1)
0	Disables segment output
1	Enables segment output

Cautions 1. SEGEN can be written only once after reset release.

2. Be sure to clear bits 2 to 7 “0”.

The segment output only pins operated by SEGEN1, SEGEN0 are as follows.

SEGEN register	Segment output only pins
	78K0R/LG3-M
SEGE1	SEG12 to SEG14 pins
SEGE0	SEG8 to SEG11 pins

(7) Input switch control register (ISC)

The segment output pins to be used alternatively with the TI04, TI02, and RxD3 pins are internally connected with a Schmitt trigger buffer. To use these pins as segment outputs, input to the Schmitt trigger buffer must be disabled, in order to prevent through-currents from entering.

ISC is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets ISC to 00H.

Figure 15-8. Format of Input Switch Control Register (ISC)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
ISC	0	0	0	ISC4	ISC3	ISC2	ISC1	ISC0

ISC4	TI04/SEG36/P53 schmitt trigger buffer control
0	Disables input
1	Enables input

ISC3	TI02/SEG37/P52 schmitt trigger buffer control
0	Disables input
1	Enables input

ISC2	RxD3/SEG39/P50 schmitt trigger buffer control
0	Disables input
1	Enables input

Caution Be sure to clear bits 7 to 5 to "0".

Remark Bits 0 and 1 of ISC are not used with the LCD controller driver.

To use the TI04/SEG36/P53, TI02/SEG37/P52, and RxD3/SEG39/P50 pins, set the PF5L and ISCn (n = 2 to 4) bits as follows, according to the function to be used.

PF5L	ISCn	Pin function
0	0	Port output (default)
0	1	Port input, timer input, or serial data input
1	0	Segment output
1	1	Setting prohibited

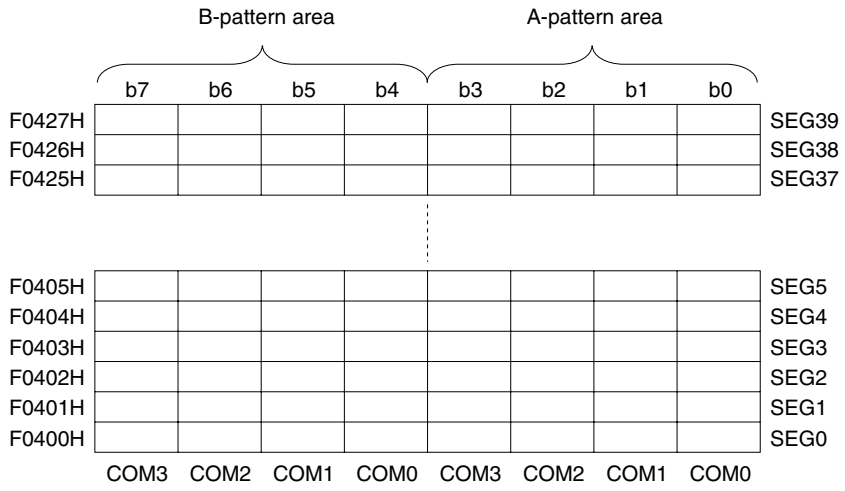
15.4 LCD Display Data Memory

The LCD display data memory is mapped at addresses F0400H to F0427H. Data in the LCD display data memory can be displayed on the LCD panel using the LCD controller/driver.

Figure 15-9 shows the relationship between the contents of the LCD display data memory and the segment/common outputs.

The areas not to be used for display can be used as normal RAM.

Figure 15-9. Relationship Between LCD Display Data Memory Contents and Segment/Common Outputs (Static, 2-time-slice, 3-time-slice, and 4-time-slice)



To use the LCD display data memory 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 memory become an A-pattern area and a B-pattern area, respectively.

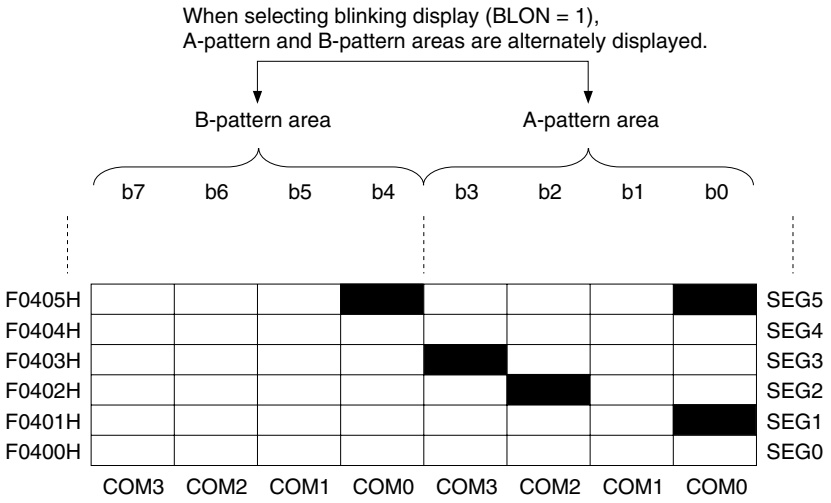
The correspondences between A-pattern area data and COM signals are as follows: bit 0 ⇔ COM0, bit 1 ⇔ COM1, bit 2 ⇔ COM2, and bit 3 ⇔ COM3.

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.

When BLON = 1 has been selected, A-pattern and B-pattern areas will be alternately displayed, according to the constant-period interrupt (INTRTC) timing of the real-time counter 2 (RTC2).

Figure 15-10. Example of Display Data When Blinking Display Has Been Selected



15.5 Setting LCD Controller/Driver

Set the LCD controller/driver using the following procedure.

(1) External resistance division method

<1> Set the external resistance division method via the MDSET0 and MDSET1 bits (bits 4 and 5 of the LCDMD register) (MDSET0 = MDSET1 = 0).

<2> To use segment output only pins, use the SEGEN register to enable segment output to them.
To use segment output pins, which are alternatively used with port pins, use the PFALL register to set them to segment output. In addition, to use the segment output pins, which are alternatively used with the TI04, TI02, and RxD3 pins, use the ISC register to disable input to the Schmitt trigger buffer.

<3> Set the display data in LCD display RAM.

<4> Set the number of time slices and the bias mode via the LCDM0 to LCDM2 bits (bits 0 to 2 of the LCDM register).
- When setting static, 2-time-slice, 3-time-slice, or 4-time-slice

<5> Select the display data area via the LCDSEL and BLON bits (bits 3 and 4 of the LCDM register).

<6> Set the LCD source clock and LCD clock via the LCDC0 register.

<7> Set (SCOC = 1) the SCOC bit (bit 6 of the LCDM register).
Non-selected waveforms are output from all the segment and common pins, and the non-display status is entered.

<8> Start output corresponding to each data memory by setting (LCDON = 1) the LCDON bit (bit 7 of the LCDM register).

(2) Internal voltage boosting method

<1> Set the internal voltage boosting method via the MDSET0 and MDSET1 bits (bits 4 and 5 of the LCDMD register) (MDSET0 = 1, MDSET1 = 0).

<2> To use segment output only pins, use the SEGEN register to enable segment output to them.

<R> Set PM00 = 1 and PM01 = 1 to use the CAPH and CAPL pins.

To use segment output pins, which are alternatively used with port pins, use the PFALL register to set them to segment output. In addition, to use the segment output pins, which are alternatively used with the TI04, TI02, and RxD3 pins, use the ISC register to disable input to the Schmitt trigger buffer.

<3> Set the display data in LCD display RAM.

<4> Set the number of time slices and the bias mode via the LCDM0 to LCDM2 bits (bits 0 to 2 of the LCDM register).

- When setting static, 2-time-slice, 3-time-slice, or 4-time-slice

(Only 1/3 bias mode can be set for the internal voltage boost method.)

<5> Select the display data area via the LCDSEL and BLON bits (bits 3 and 4 of the LCDM register).

<6> Set the LCD source clock and LCD clock via the LCDC0 register.

<7> Set the reference voltage (adjust the contrast) via the VLCD register.

<8> Wait for the reference voltage setup time (2 ms (min.)) after setting of the VLCD register.

<9> Set (VLCON = 1) the VLCON bit (bit 5 of the LCDM register) to start the voltage boost circuit operation.

<10> Wait for the voltage boost wait time after setting of VLCON (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS**).

<11> Set (SCOC = 1) the SCOC bit (bit 6 of the LCDM register).

Non-selected waveforms are output from all the segment and common pins, and the non-display status is entered.

<12> Start output corresponding to each data memory by setting (LCDON = 1) the LCDON bit (bit 7 of the LCDM register).

Caution When stopping the operation of the voltage boost circuit, be sure to set SCOC and LCDON to 0 before setting VLCON to 0.

(3) Capacitor split method

<1> Set the capacitor split method via the MDSET0 and MDSET1 bits (bits 4 and 5 of the LCDMD register) (MDSET0 = 0, MDSET1 = 1).

<2> To use segment output only pins, use the SEGEN register to enable segment output to them.

<R> Set PM00 = 1 and PM01 = 1 to use the CAPH and CAPL pins.

To use segment output pins, which are alternatively used with port pins, use the PFALL register to set them to segment output. In addition, to use the segment output pins, which are alternatively used with the TI04, TI02, and RxD3 pins, use the ISC register to disable input to the Schmitt trigger buffer.

<3> Set the display data in LCD display RAM.

<4> Set the number of time slices and the bias mode via the LCDM0 to LCDM2 bits (bits 0 to 2 of the LCDM register).

(Only 1/3 bias mode can be set for the capacitor split method)

<5> Select the display data area via the LCDSEL and BLON bits (bits 3 and 4 of the LCDM register).

<6> Set the LCD source clock and LCD clock via the LCDC0 register.

<7> Set (VLCON = 1) the VLCON bit (bit 5 of the LCDM register) to start the voltage reduction circuit operation.

<8> Wait for the voltage capacitor split wait time after setting of VLCON (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS**).

<9> Set (SCOC = 1) the SCOC bit (bit 6 of the LCDM register).

Non-selected waveforms are output from all the segment and common pins, and the non-display status is entered.

<10> Start output corresponding to each data memory by setting (LCDON = 1) the LCDON bit (bit 7 of the LCDM register).

Caution When stopping the operation of the capacitor split circuit, be sure to set SCOC and LCDON to 0 before setting VLCON to 0.

15.6 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 15-3. 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.

Table 15-3. COM Signals

COM Signal / Number of Time Slices	COM0	COM1	COM2	COM3
Static display mode				
Two-time-slice mode			Open	Open
Three-time-slice mode				Open
Four-time-slice mode				

(2) Segment signals

The segment signals correspond to the LCD display data memory (refer to **15.4 LCD Display Data Memory**).

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 SEG39).

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 memory, and write the bit data that corresponds to the desired display pattern on a one-to-one basis.

LCD display data memory bits 1 to 3, bits 2 and 3, and bit 3 are not used for LCD display in the static display, two-time slot, and three-time slot, respectively. So these bits can be used for purposes other than display.

(3) Output waveforms of common and segment signals

The voltages listed in Table 15-4 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 15-4. LCD Drive Voltage

(a) Static display mode

Segment Signal / Common Signal	Select Signal Level	Deselect Signal Level
	V_{SS}/V_{LC0}	V_{LC0}/V_{SS}
V_{LC0}/V_{SS}	$-V_{LCD}/+V_{LCD}$	0 V/0 V

(b) 1/2 bias method

Segment Signal / Common Signal		Select Signal Level	Deselect Signal Level
		V_{SS}/V_{LC0}	V_{LC0}/V_{SS}
Select signal level	V_{LC0}/V_{SS}	$-V_{LCD}/+V_{LCD}$	0 V/0 V
Deselect signal level	$V_{LC1} = V_{LC2}$	$-\frac{1}{2}V_{LCD}/+\frac{1}{2}V_{LCD}$	$+\frac{1}{2}V_{LCD}/-\frac{1}{2}V_{LCD}$

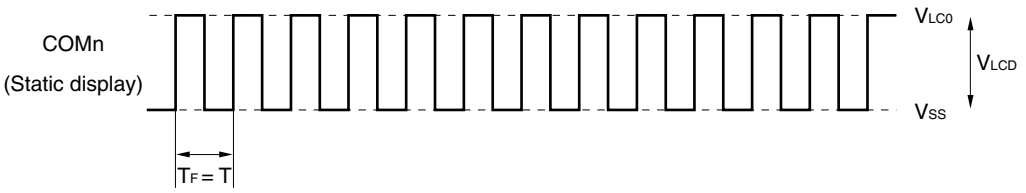
(c) 1/3 bias method

Segment Signal / Common Signal		Select Signal Level	Deselect Signal Level
		V_{SS}/V_{LC0}	V_{LC1}/V_{LC2}
Select signal level	V_{LC0}/V_{SS}	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$
Deselect signal level	V_{LC2}/V_{LC1}	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$	$+\frac{1}{3}V_{LCD}/-\frac{1}{3}V_{LCD}$

Figure 15-11 shows the common signal waveforms, and Figure 15-12 shows the voltages and phases of the common and segment signals.

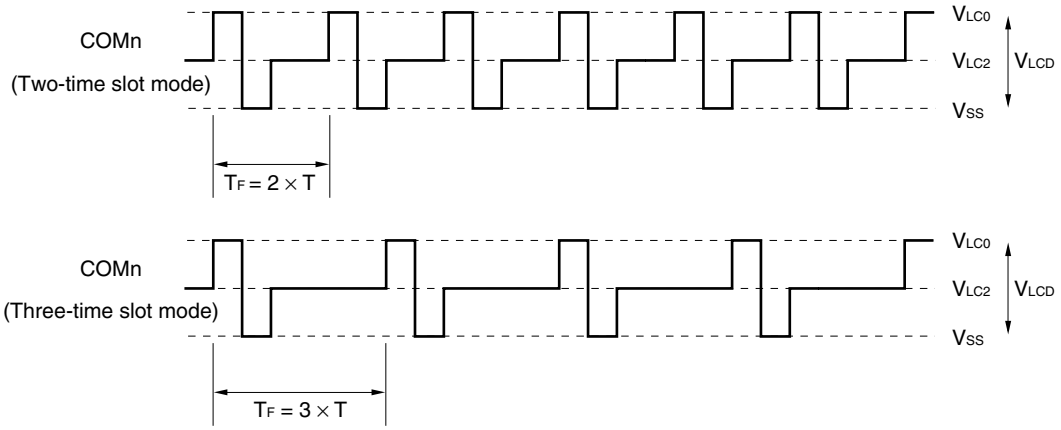
Figure 15-11. Common Signal Waveforms

(a) Static display mode



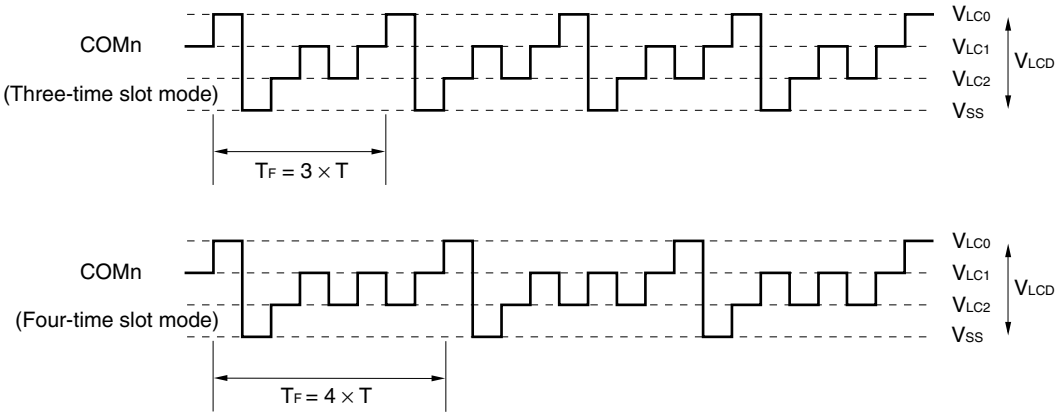
T: One LCD clock period T_F: Frame frequency

(b) 1/2 bias method



T: One LCD clock period T_F: Frame frequency

(c) 1/3 bias method



T: One LCD clock period T_F: Frame frequency

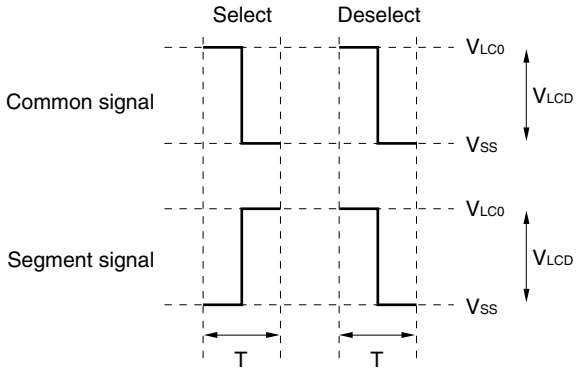
< Example of calculation of LCD frame frequency (When four-time slot mode is used) >

LCD clock: $32768/2^8$ = 256 Hz (When setting to LCDC0 = 04H)

LCD frame frequency: 64 Hz

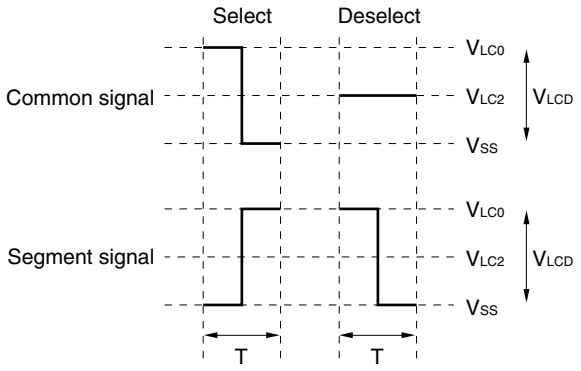
Figure 15-12. Voltages and Phases of Common and Segment Signals

(a) Static display mode



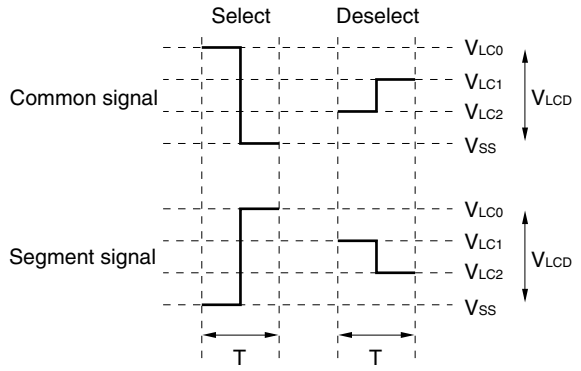
T: One LCD clock period

(b) 1/2 bias method



T: One LCD clock period

(c) 1/3 bias method



T: One LCD clock period

15.7 Display Modes

15.7.1 Static display example

Figure 15-14 shows how the three-digit LCD panel having the display pattern shown in Figure 15-13 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 memory (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 15-5 at the timing of the common signal COM0; see Figure 15-13 for the relationship between the segment signals and LCD segments.

Table 15-5. Select and Deselect Voltages (COM0)

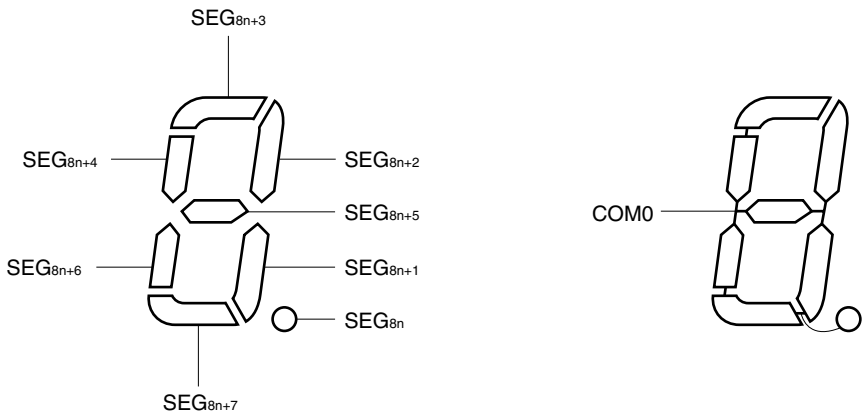
Segment / Common	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
COM0	Select	Deselect	Select	Select	Deselect	Select	Select	Select

According to Table 15-5, it is determined that the bit-0 pattern of the display data memory locations (F0408H to F040FH) must be 10110111.

Figure 15-15 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, $+V_{LCD}/-V_{LCD}$, 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 15-13. Static LCD Display Pattern and Electrode Connections



Remark n = 0 to 4

Figure 15-14. Example of Connecting Static LCD Panel

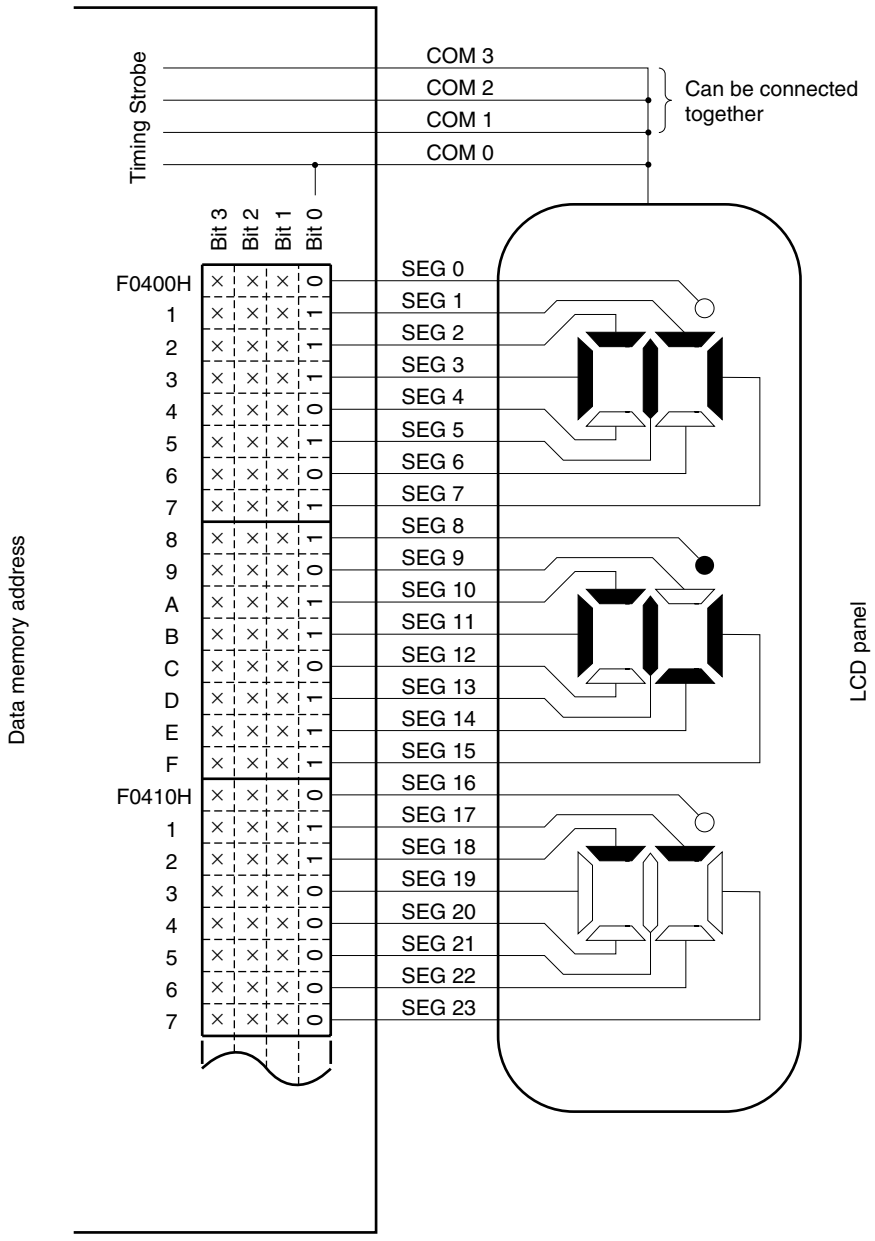
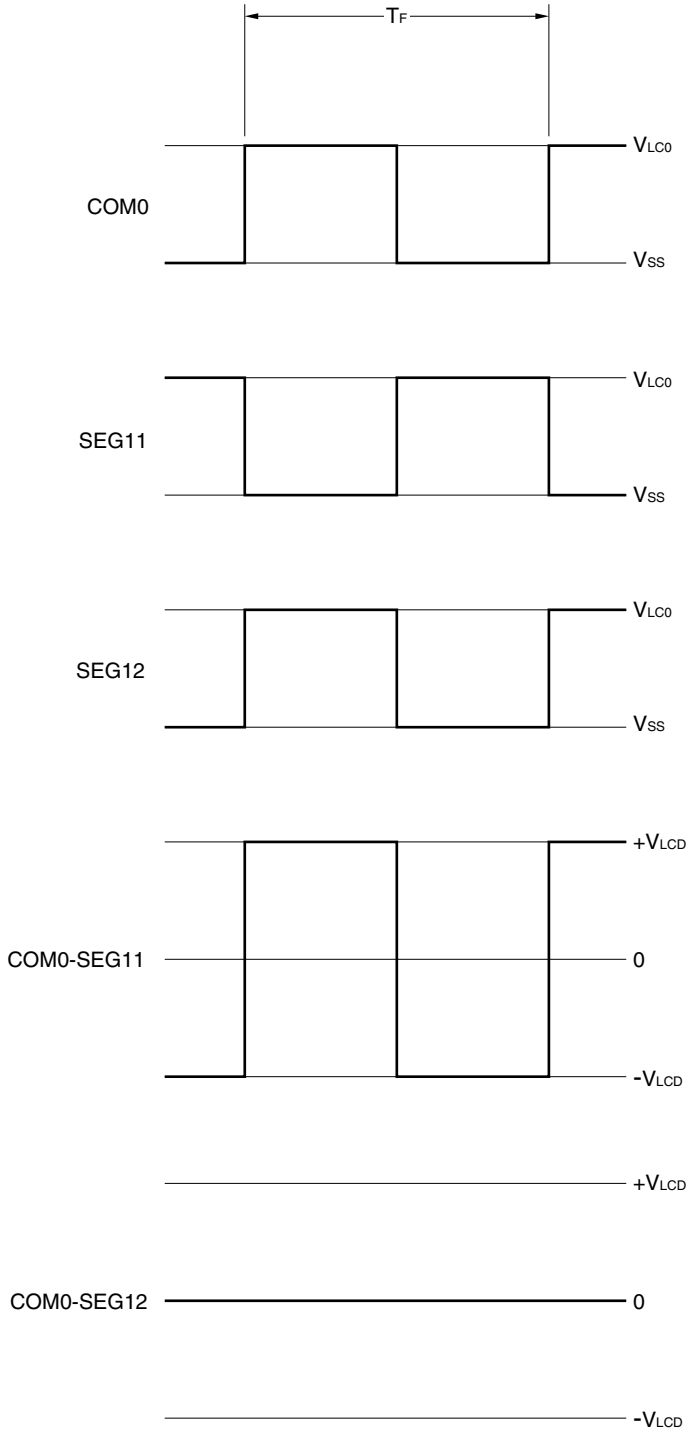


Figure 15-15. Static LCD Drive Waveform Examples



15.7.2 Two-time-slice display example

Figure 15-17 shows how the 6-digit LCD panel having the display pattern shown in Figure 15-16 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 memory (F0400H to F0417H) correspond to this display.

The following description focuses on numeral "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 15-6 at the timing of the common signals COM0 and COM1; see Figure 15-16 for the relationship between the segment signals and LCD segments.

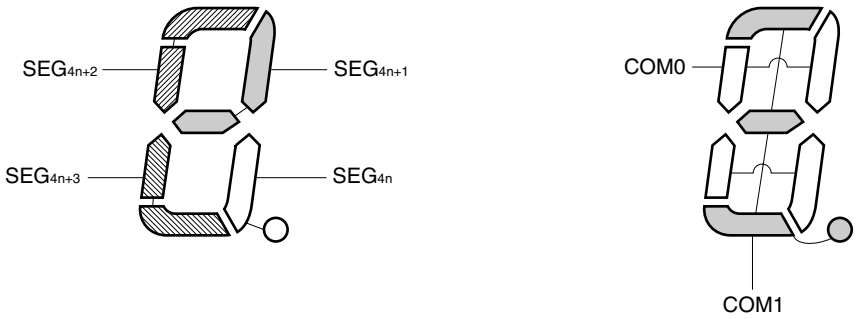
Table 15-6. Select and Deselect Voltages (COM0 and COM1)

Segment / Common	SEG12	SEG13	SEG14	SEG15
COM0	Select	Select	Deselect	Deselect
COM1	Deselect	Select	Select	Select

According to Table 15-6, it is determined that the display data memory location (F040FH) that corresponds to SEG15 must contain xx10.

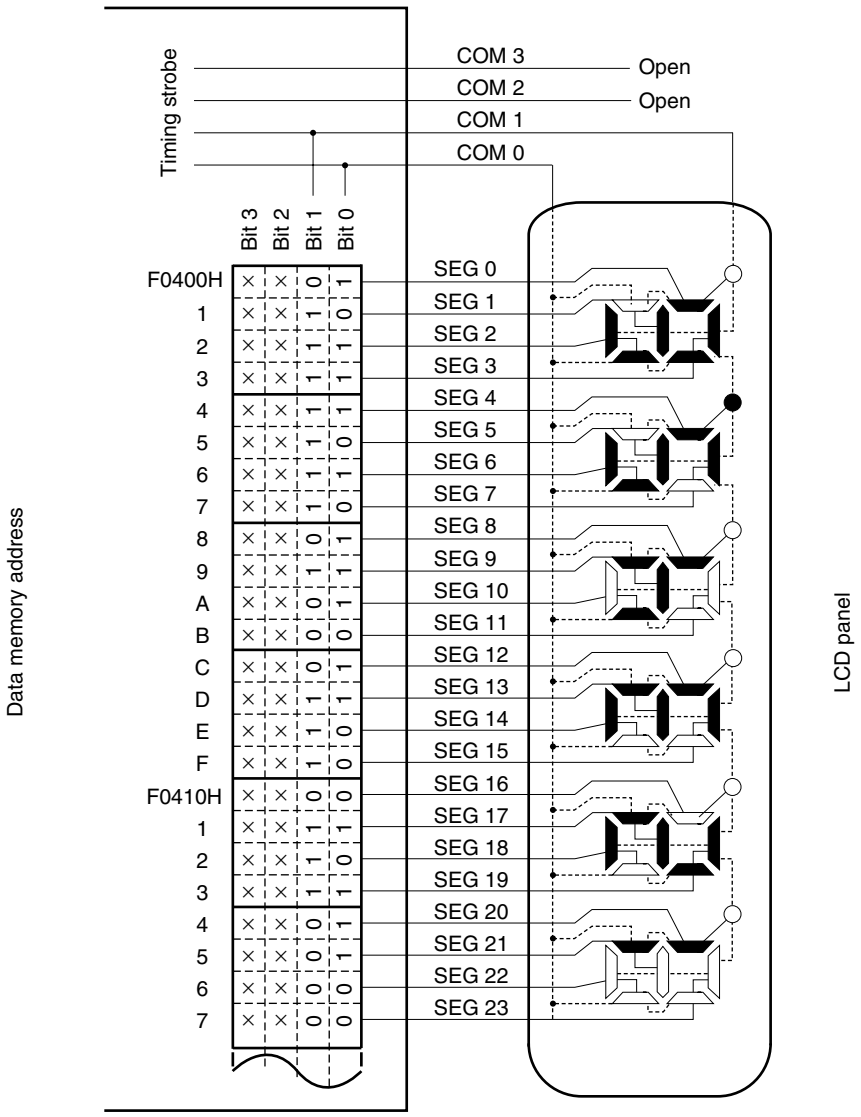
Figure 15-18 shows examples of LCD drive waveforms between the SEG15 signal and each common signal. When the select voltage is applied to SEG15 at the timing of COM1, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 15-16. Two-Time-Slice LCD Display Pattern and Electrode Connections



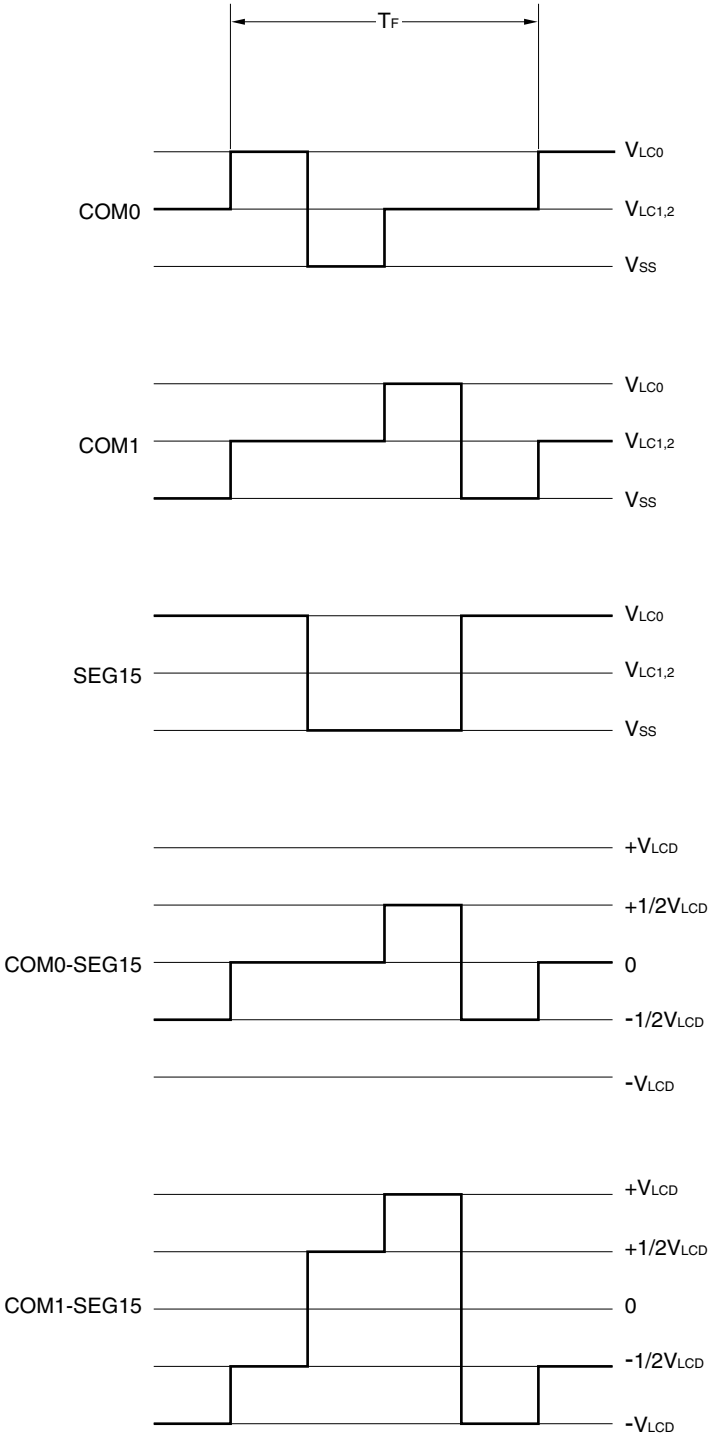
Remark n = 0 to 9

Figure 15-17. Example of Connecting Two-Time-Slice LCD Panel



×: Can always be used to store any data because the two-time-slice mode is being used.

Figure 15-18. Two-Time-Slice LCD Drive Waveform Examples (1/2 Bias Method)



15.7.3 Three-time-slice display example

Figure 15-20 shows how the 8-digit LCD panel having the display pattern shown in Figure 15-19 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 memory (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 15-7 at the timing of the common signals COM0 to COM2; see Figure 15-19 for the relationship between the segment signals and LCD segments.

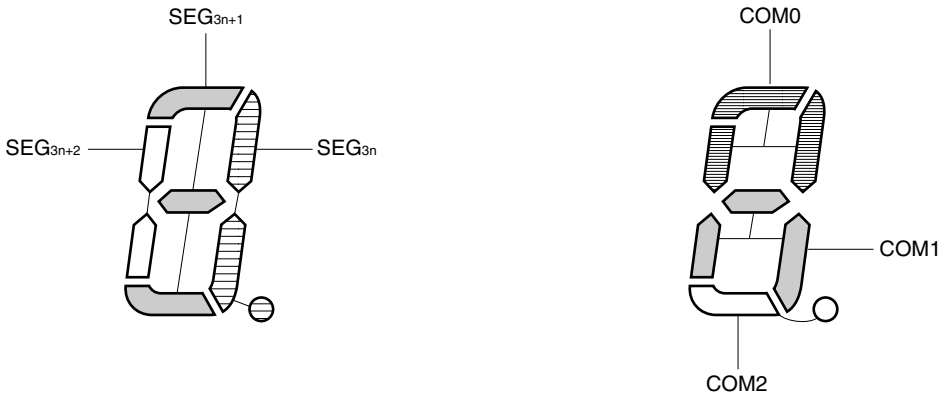
Table 15-7. Select and Deselect Voltages (COM0 to COM2)

Segment / Common	SEG6	SEG7	SEG8
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	–

According to Table 15-7, it is determined that the display data memory location (F0406H) that corresponds to SEG6 must contain x110.

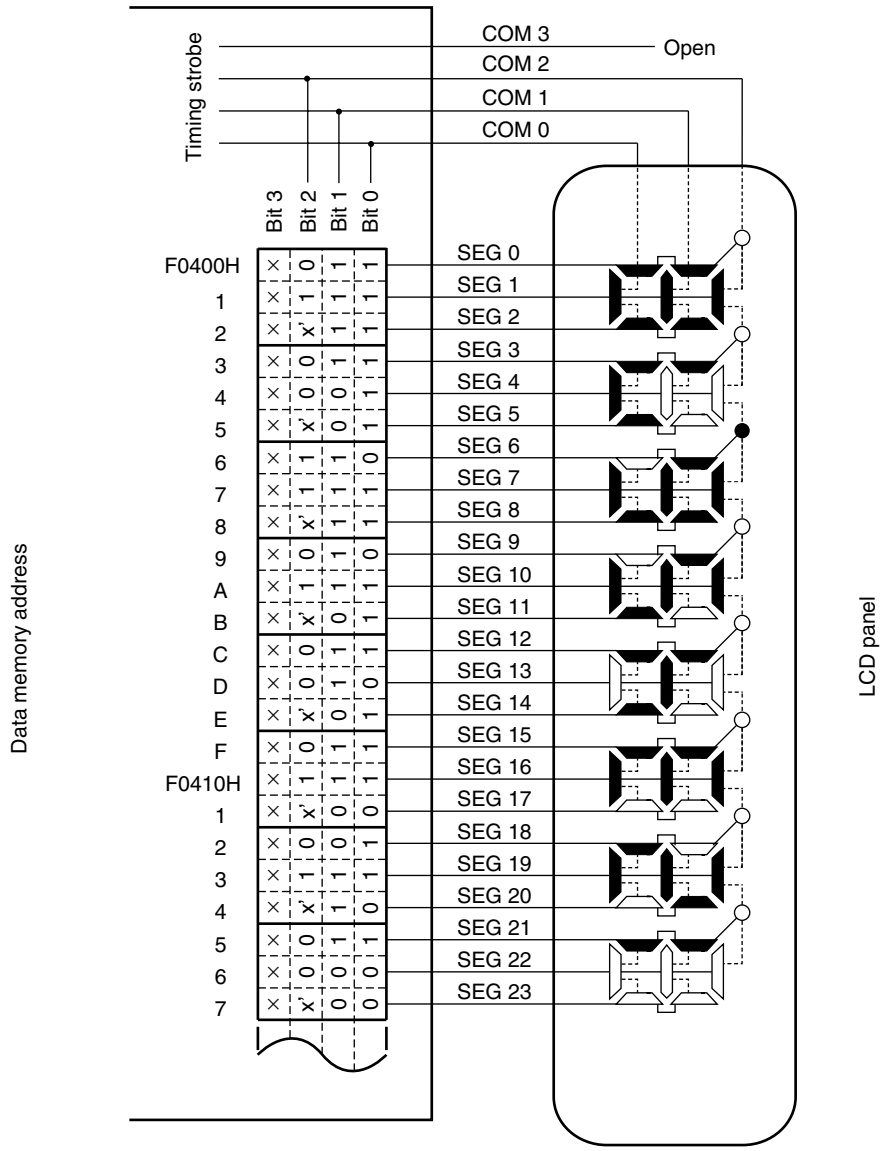
Figures 15-21 and 15-22 show examples of LCD drive waveforms between the SEG6 signal and each common signal in the 1/2 and 1/3 bias methods, respectively. When the select voltage is applied to SEG6 at the timing of COM1 or COM2, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 15-19. Three-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 12

Figure 15-20. Example of Connecting Three-Time-Slice LCD Panel



×': Can be used to store any data because there is no corresponding segment in the LCD panel.

×: Can always be used to store any data because the three-time-slice mode is being used.

Figure 15-21. Three-Time-Slice LCD Drive Waveform Examples (1/2 Bias Method)

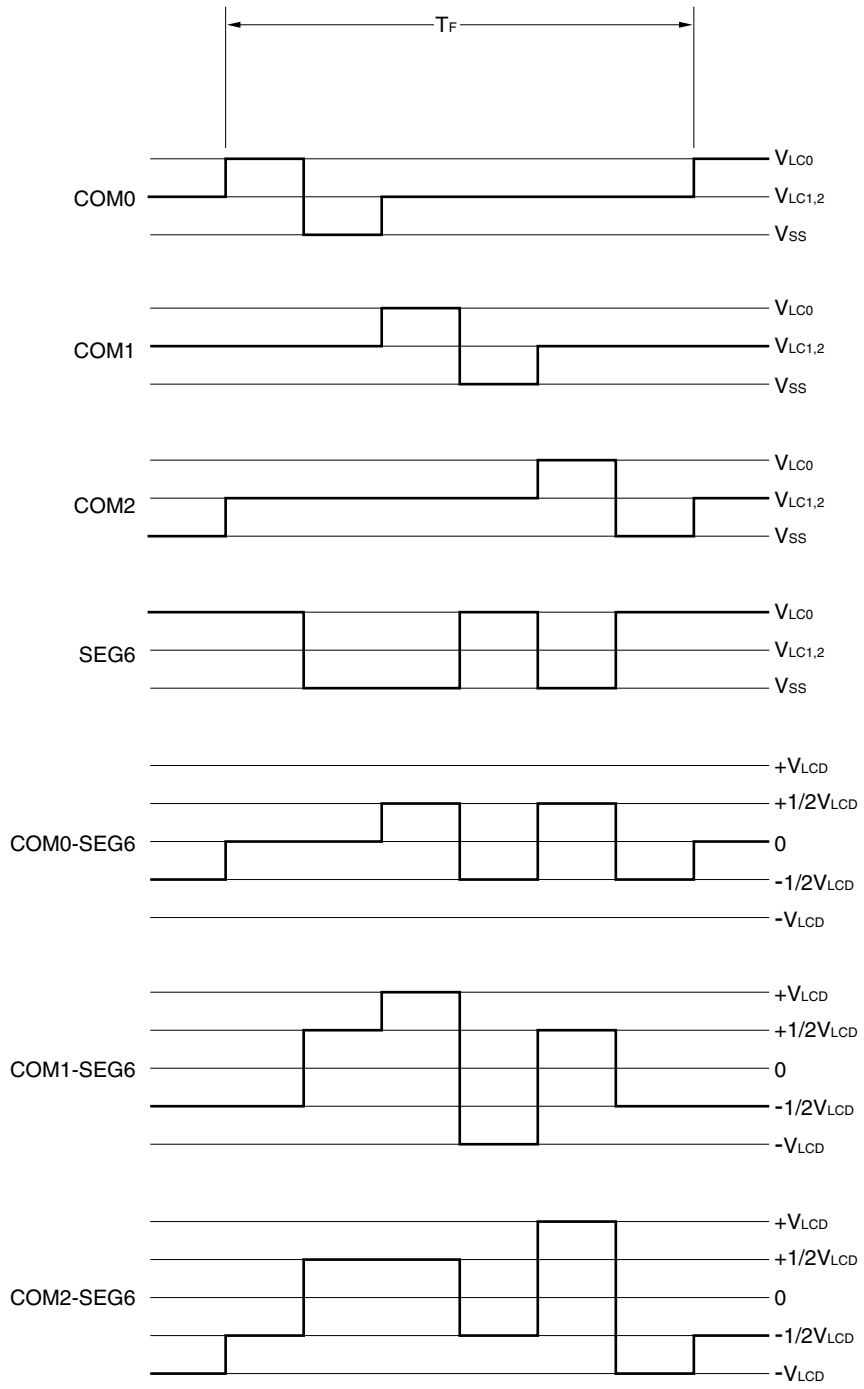
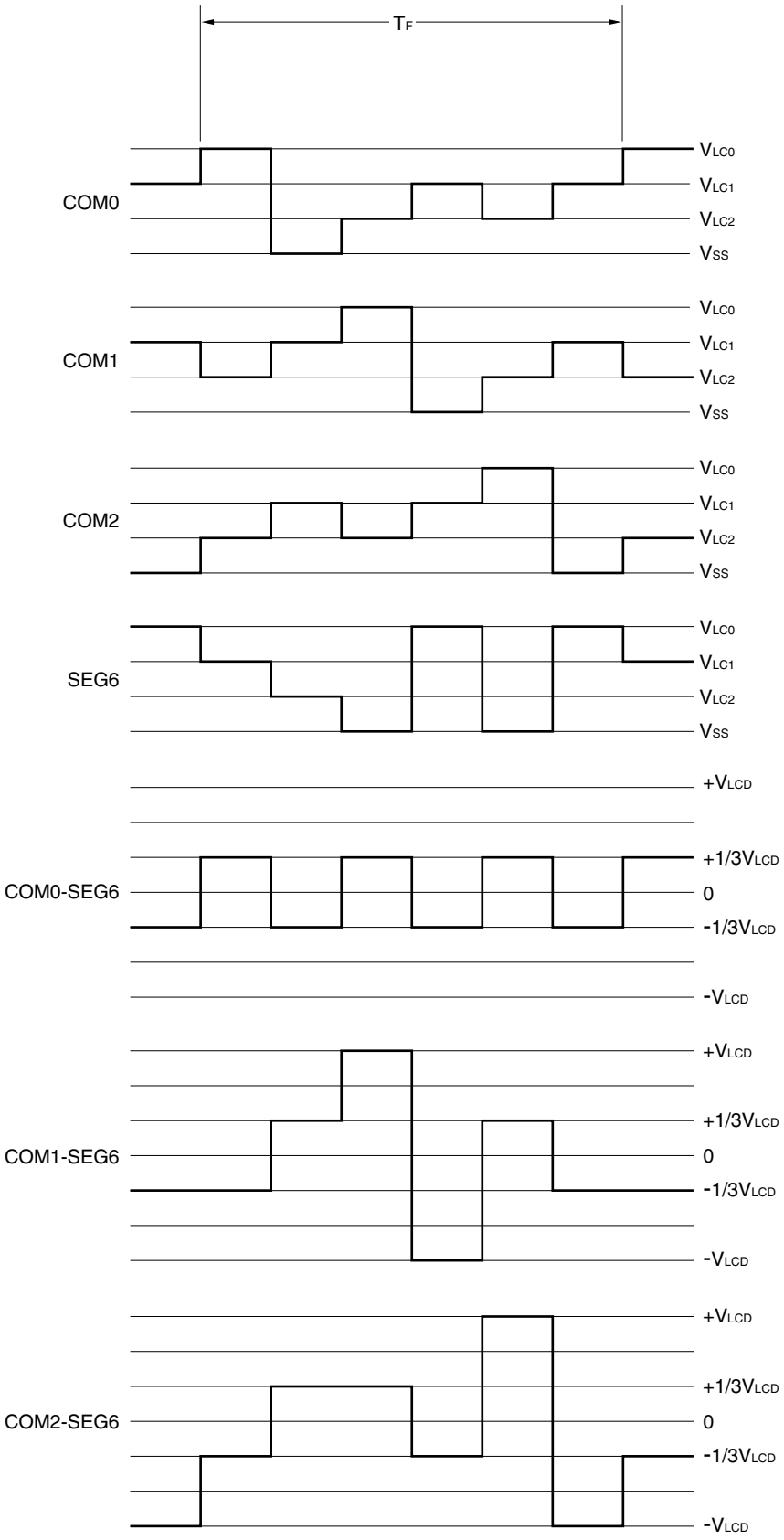


Figure 15-22. Three-Time-Slice LCD Drive Waveform Examples (1/3 Bias Method)



15.7.4 Four-time-slice display example

Figure 15-24 shows how the 12-digit LCD panel having the display pattern shown in Figure 15-23 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 memory (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 Table 15-8 at the timing of the common signals COM0 to COM3; see Figure 15-23 for the relationship between the segment signals and LCD segments.

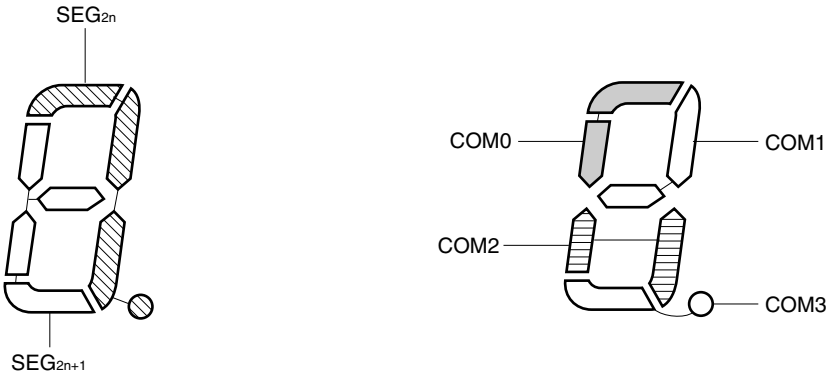
Table 15-8. Select and Deselect Voltages (COM0 to COM3)

Segment / Common	SEG12	SEG13
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to Table 15-8, it is determined that the display data memory location (F040CH) that corresponds to SEG12 must contain 1101.

Figure 15-25 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, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 15-23. Four-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 19

Figure 15-24. Example of Connecting Four-Time-Slice LCD Panel

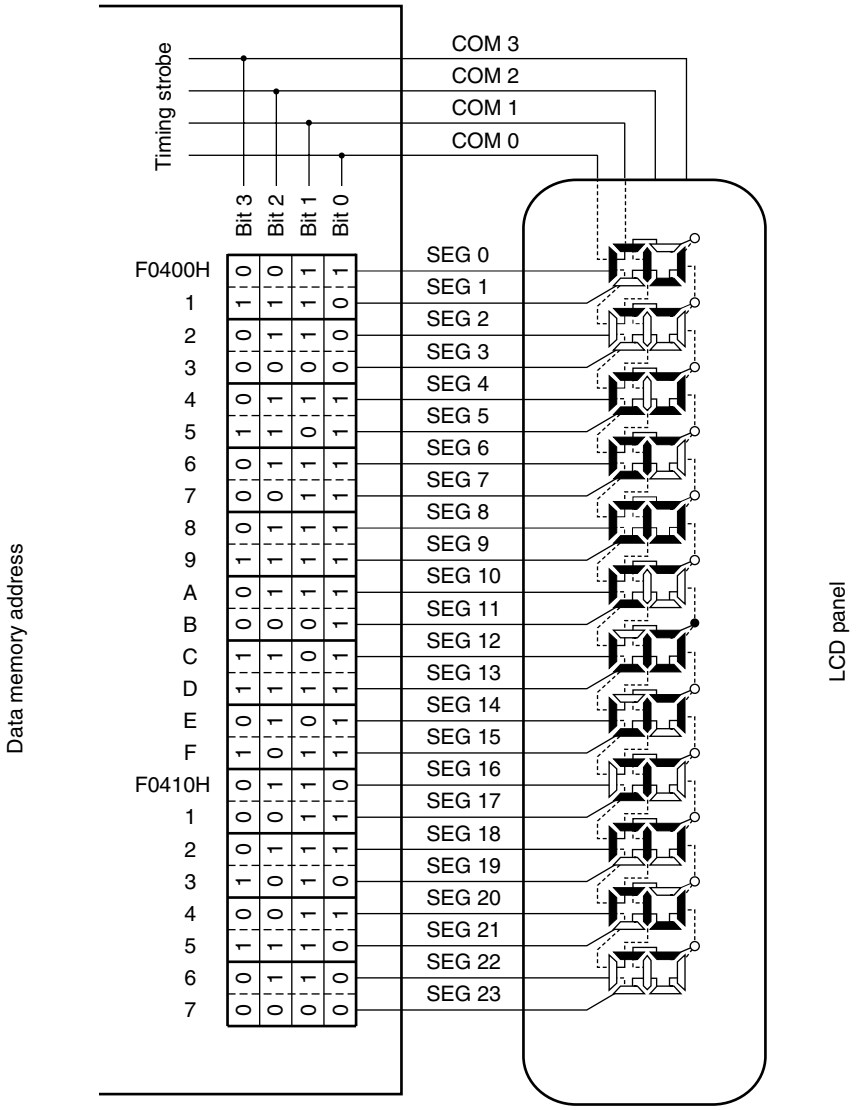
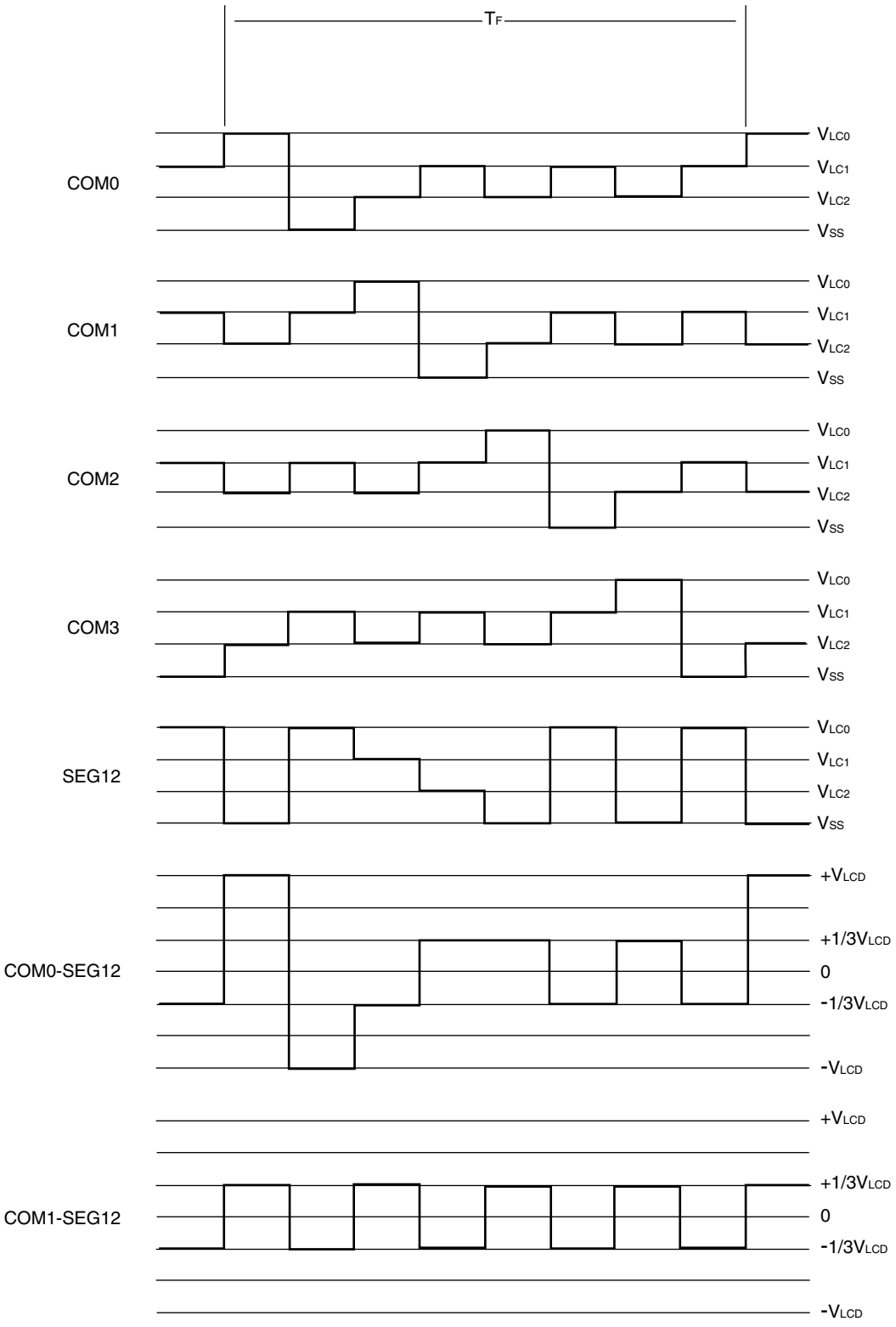


Figure 15-25. Four-Time-Slice LCD Drive Waveform Examples (1/3 Bias Method)



Remark The waveforms for COM2-SEG12 and COM3-SEG12 are omitted.

15.8 Supplying LCD Drive Voltages V_{LC0}, V_{LC1}, and V_{LC2}

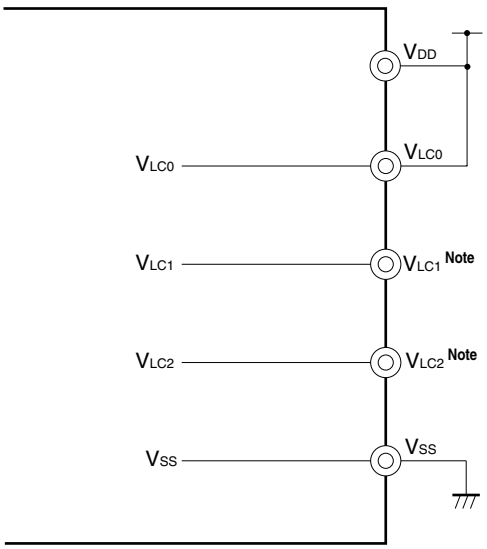
With the 78K0R/Lx3-M microcontrollers, a LCD drive power supply can be generated using either of three types of methods: external resistance division method, internal voltage boosting method, or capacitor split method.

15.8.1 External resistance division method

The 78K0R/Lx3-M microcontrollers can also use external voltage divider resistors for generating LCD drive power supplies, without using internal resistors. Figure 15-26 shows examples of LCD drive voltage connection, corresponding to each bias method.

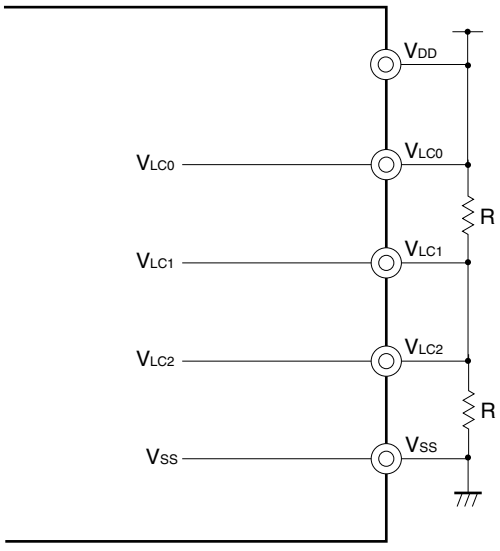
Figure 15-26. Examples of LCD Drive Power Connections (External Resistance Division Method) (1/2)

(a) Static display mode



$V_{LC0} = V_{DD}$

(b) 1/2 bias method



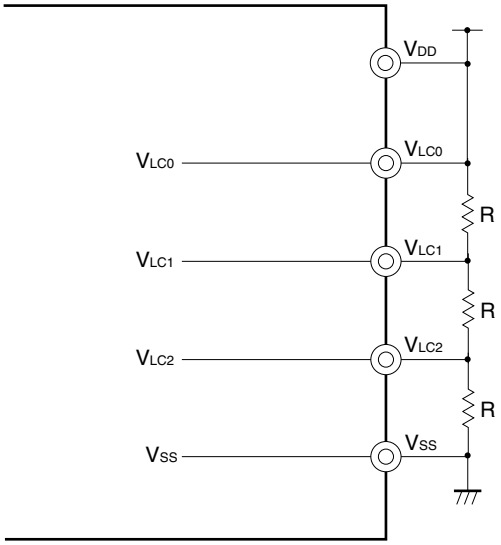
$V_{LC0} = V_{DD}$

Note Connect V_{LC1} and V_{LC2} directly to GND or V_{LC0}.

Caution To stabilize the potential of the V_{LC0} to V_{LC2} pins, it is recommended to connect a capacitor of about 0.1 μF between each of the pins from V_{LC0} to V_{LC2} and the GND pin as needed.

Figure 15-26. Examples of LCD Drive Power Connections (External Resistance Division Method) (2/2)

(c) 1/3 bias method



$V_{LC0} = V_{DD}$

Caution To stabilize the potential of the V_{LC0} to V_{LC2} pins, it is recommended to connect a capacitor of about 0.1 μF between each of the pins from V_{LC0} to V_{LC2} and the GND pin as needed.

15.8.2 Internal voltage boosting method

The 78K0R/Lx3-M microcontrollers contain an internal voltage boost circuit for generating LCD drive power supplies. The internal voltage boost circuit and external capacitors (0.47 μF ±30%) are used to generate an LCD drive voltage. Only 1/3 bias mode can be set for the internal voltage boost method.

The LCD drive voltage of the internal voltage boost method can supply a constant voltage, regardless of changes in V_{DD}, because it is a power supply separate from the main unit.

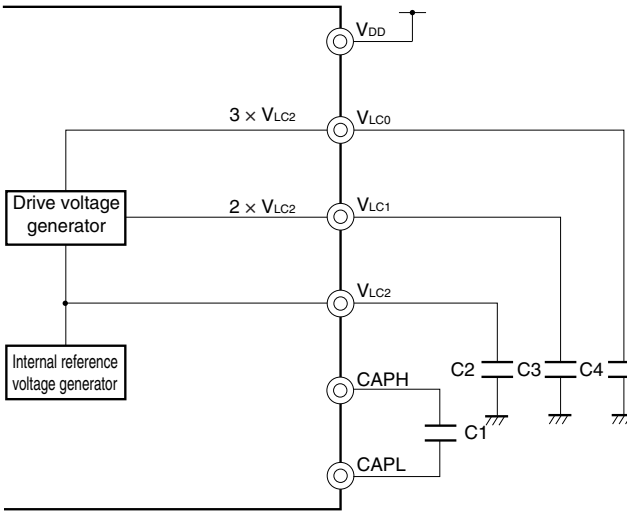
In addition, a contrast can be adjusted by using the LCD boost level control register (VLCD).

Table 15-9. LCD Drive Voltages (Internal Voltage Boosting Method)

Bias Method / LCD Drive Voltage Pin	1/3 Bias Method
V_{LC0}	3 x V_{LC2}
V_{LC1}	2 x V_{LC2}
V_{LC2}	LCD reference voltage

Figure 15-27. Examples of LCD Drive Power Connections (Internal Voltage Boosting Method)

(a) 1/3 bias method



Remark Use a capacitor with as little leakage as possible.
In addition, make C1 a nonpolar capacitor.

15.8.3 Capacitor split method

The 78K0R/Lx3-M microcontrollers contain an internal voltage reduction circuit for generating LCD drive power supplies. The internal voltage reduction circuit and external capacitors (0.47 μF ±30%) are used to generate an LCD drive voltage. Only 1/3 bias mode can be set for the capacitor split method.

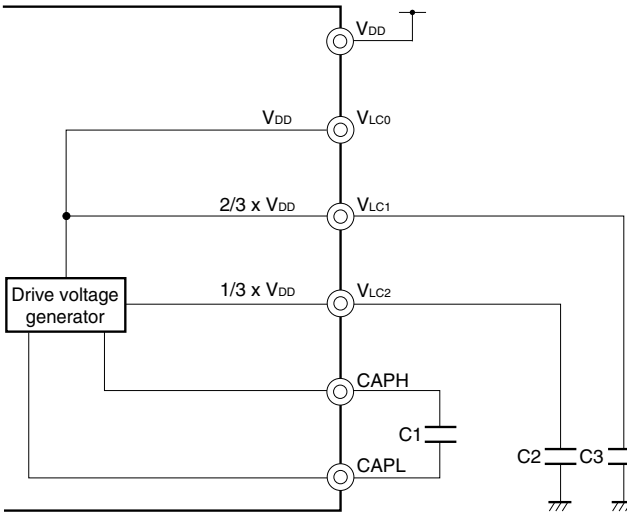
Different from the external resistance division method, there is always no current flowing with the capacitor split method, so current consumption can be reduced.

<R> For the capacitor split method, set PM00 = 1 and PM01 = 1 to use the CAPH and CAPL pins.

Table 15-10. LCD Drive Voltages (Capacitor Split Method)

Bias Method / LCD Drive Voltage Pin	1/3 Bias Method
V_{LC0}	V_{DD}
V_{LC1}	2/3 x V_{DD}
V_{LC2}	1/3 x V_{DD}

Figure 15-28. Example of LCD Drive Power Connections (Capacitor Split Method)
• 1/3 bias method



Remark Use a capacitor with as little leakage as possible.
In addition, make C1 a nonpolar capacitor.

15.9 Selection of LCD Display Data

With the 78K0R/Lx3-M microcontroller, to use the LCD display data memory when the number of time slices is static, two, three, or four, the LCD display data 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 memory)
- Displaying a B-pattern area data (higher four bits of LCD display data memory)
- Alternately displaying A-pattern and B-pattern area data (blinking display corresponding to the constant-period interrupt timing of the real-time counter 2 (RTC2))

15.9.1 A-pattern area and B-pattern area data display

When BLON = LCDSEL = 0, A-pattern area (lower four bits of the LCD display data memory) data will be output as the LCD display data.

When BLON = 0, and LCDSEL = 1, B-pattern area (higher four bits of the LCD display data memory) data will be output as the LCD display data.

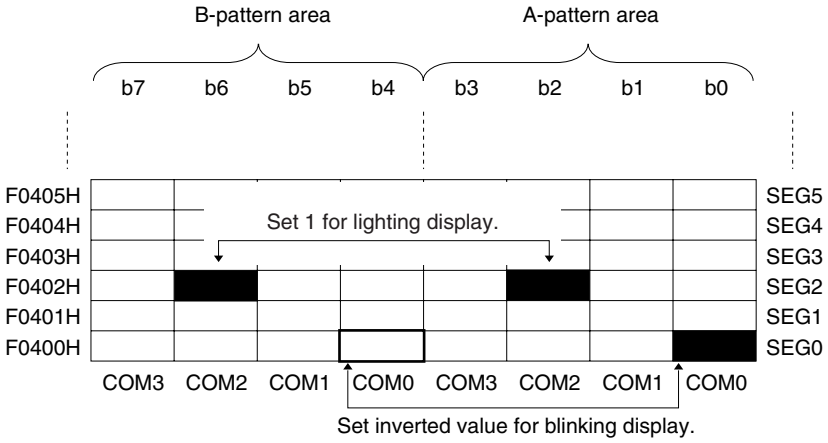
Refer to **15.4 LCD Display Data Memory** about the display area.

15.9.2 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 the real-time counter 2 (RTC2). Refer to **CHAPTER 8 REAL-TIME COUNTER 2** about the setting of the RTC2 constant-period interrupt (INTRTC) timing.

For blinking display of the LCD, set inverted values to the B-pattern area bits corresponding to the A-pattern area bits. (Example: Set 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: Set 1 to bit 2 of F0402H, and set 1 to bit 6 of F0402H for lighting display.)

Figure 15-29. Example of LCD Display Data Setting During Pattern-Switching Display



Refer to **15.4 LCD Display Data Memory** about the display area.

Next, the timing operation of display switching is shown.

Figure 15-30. Switching Operation from A-Pattern Display to Blinking Display

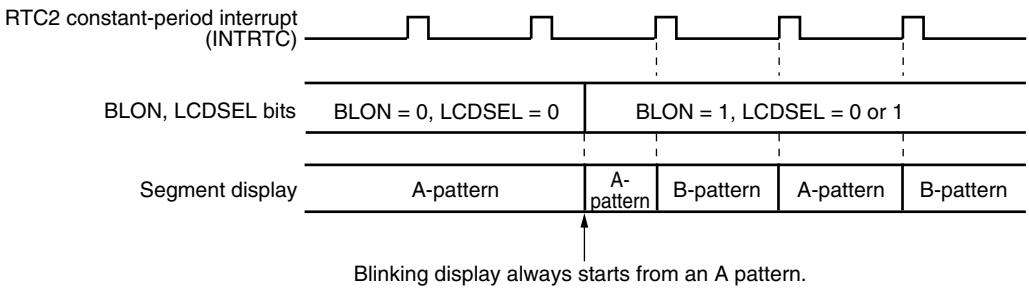
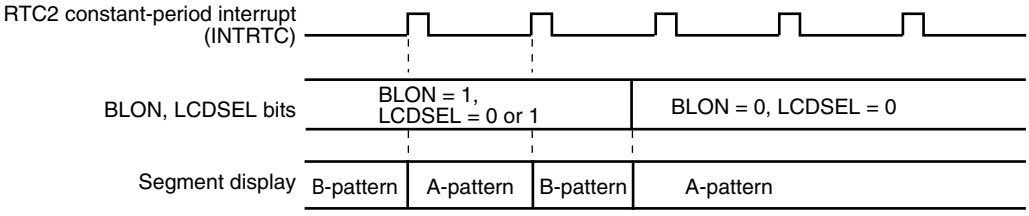


Figure 15-31. Switching Operation from Blinking Display to A-Pattern Display



CHAPTER 16 MULTIPLIER/DIVIDER

16.1 Functions of Multiplier/Divider

The multiplier/divider is mounted onto all 78K0R/Lx3-M microcontroller products.
The multiplier/divider has the following functions.

- 16 bits × 16 bits = 32 bits (multiplication)
- 32 bits ÷ 32 bits = 32 bits, 32-bit remainder (division)

16.2 Configuration of Multiplier/Divider

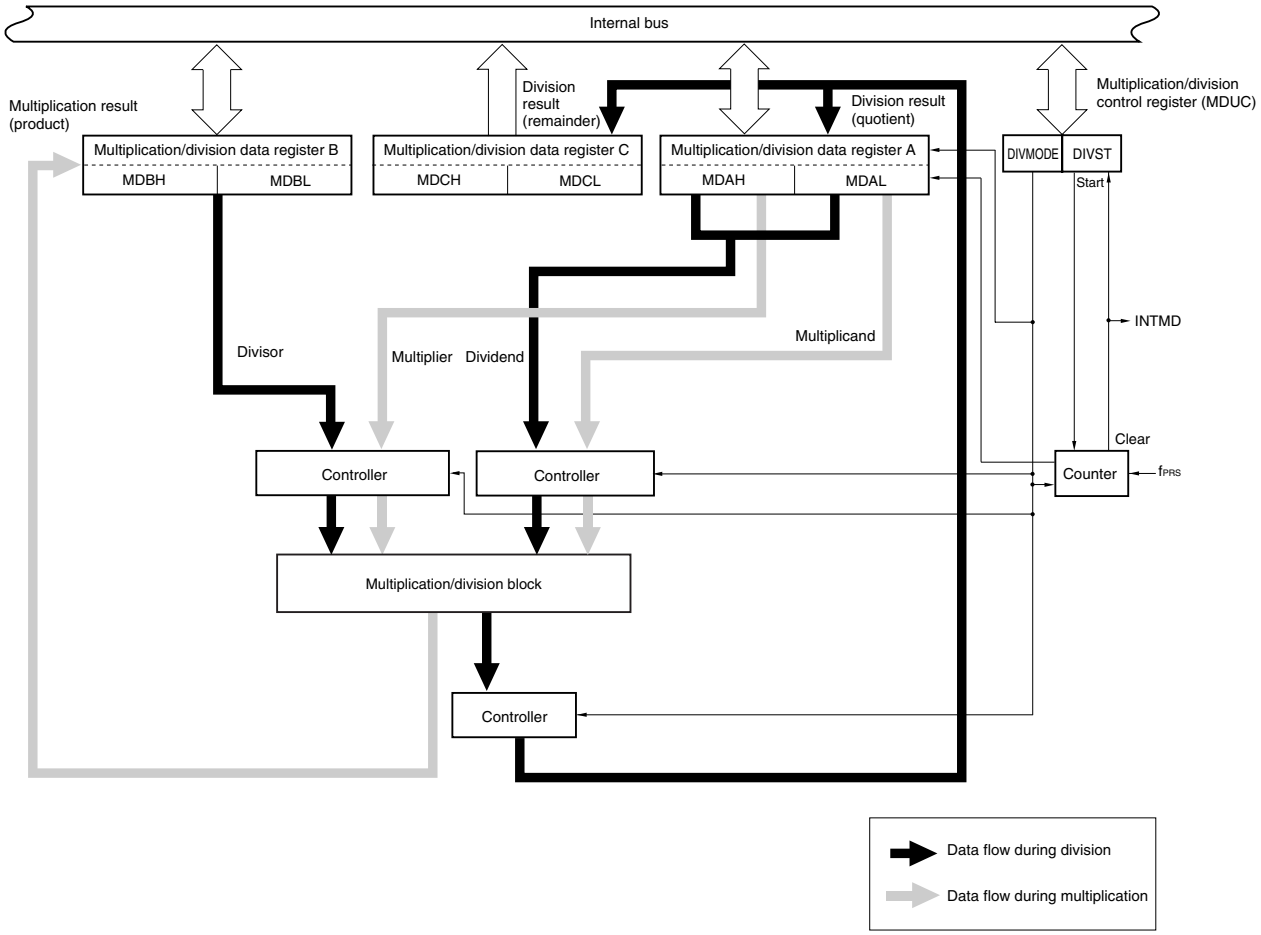
The multiplier/divider consists of the following hardware.

Table 16-1. Configuration of Multiplier/Divider

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 16-1 shows a block diagram of the multiplier/divider.

Figure 16-1. Block Diagram of Multiplier/Divider



(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, 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.

MDAH and MDAL can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 16-2. Format of Multiplication/Division Data Register A (MDAH, MDAL)

Address: FFFF0H, FFFF1H, FFFF2H, FFFF3H After reset: 0000H, 0000H R/W

Symbol FFFF3H FFFF2H

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

Symbol FFFF1H FFFF0H

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

Cautions 1. Do not rewrite the MDAH and MDAL values during division operation processing (while the multiplication/division control register (MDUC) is 81H). The operation will be executed in this case, but the operation result will be an undefined value.

2. The MDAH and MDAL values read during division operation processing (while MDUC is 81H) will not be guaranteed.

The following table shows the functions of MDAH and MDAL during operation execution.

Table 16-2. Functions of MDAH and MDAL During Operation Execution

DIVMODE	Operation Mode	Setting	Operation Result
0	Multiplication mode	MDAH: Multiplier MDAL: Multiplicand	–
1	Division mode	MDAH: Divisor (higher 16 bits) MDAL: Dividend (lower 16 bits)	MDAH: Division result (quotient) Higher 16 bits MDAL: Division result (quotient) Lower 16 bits

Remark DIVMODE: Bit 7 of the multiplication/division control register (MDUC)

(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 set the divisor data in the division mode.

MDBH and MDBL can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 16-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)

Address: FFFF4H, FFFF5H, FFFF6H, FFFF7H After reset: 0000H, 0000H R/W

Symbol	FFFF7H								FFFF6H							
MDBH	MDBH 15	MDBH 14	MDBH 13	MDBH 12	MDBH 11	MDBH 10	MDBH 9	MDBH 8	MDBH 7	MDBH 6	MDBH 5	MDBH 4	MDBH 3	MDBH 2	MDBH 1	MDBH 0

Symbol	FFFF5H								FFFF4H							
MDBL	MDBL 15	MDBL 14	MDBL 13	MDBL 12	MDBL 11	MDBL 10	MDBL 9	MDBL 8	MDBL 7	MDBL 6	MDBL 5	MDBL 4	MDBHL 3	MDBL 2	MDBL 1	MDBL 0

Cautions 1. Do not rewrite the MDBH and MDBL values during division operation processing (while the multiplication/division control register (MDUC) is 81H). The operation result will be an undefined value.

2. Do not set MDBH and MDBL to 0000H in the division mode. If they are set, the operation result will be an undefined value.

The following table shows the functions of MDBH and MDBL during operation execution.

Table 16-3. Functions of MDBH and MDBL During Operation Execution

DIVMODE	Operation Mode	Setting	Operation Result
0	Multiplication mode	–	MDBH: Multiplication result (product) Higher 16 bits MDBL: Multiplication result (product) Lower 16 bits
1	Division mode	MDBH: Divisor (higher 16 bits) MDBL: Dividend (lower 16 bits)	–

Remark DIVMODE: Bit 7 of the multiplication/division control register (MDUC)

(3) Multiplication/division data register C (MDCL, MDCH)

The MDCH and MDCL registers store remainder value of the operation result in the division mode. They are not used in the multiplication mode.

MDCH and MDCL can be read by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 16-4. Format of Multiplication/Division Data Register C (MDCH, MDCL)

Address: F00E0H, F00E1H, F00E2H, F00E3H After reset: 0000H, 0000H R

Symbol F00E3H F00E2H

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

Symbol F00E1H F00E0H

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

Caution The MDCH and MDCL values read during division operation processing (while the multiplication/division control register (MDUC) is 81H) will not be guaranteed.

Table 16-4. Functions of MDCH and MDCL During Operation Execution

DIVMODE	Operation Mode	Setting	Operation Result
0	Multiplication mode	–	–
1	Division mode	–	MDCH: Remainder (higher 16 bits) MDCL: Remainder (lower 16 bits)

Remark DIVMODE: Bit 7 of the multiplication/division control register (MDUC)

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

<Multiplier A> <Multiplier B> <Product>

MDAL (bits 15 to 0) × MDAH (bits 15 to 0) = [MDBH (bits 15 to 0), MDBL (bits 15 to 0)]

- Register configuration during division

<Dividend> <Divisor>

[MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ÷ [MDBH (bits 15 to 0), MDBL (bits 15 to 0)] =

<Quotient> <Remainder>

[MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ··· [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

16.3 Register Controlling Multiplier/Divider

The multiplier/divider is controlled by using the multiplication/division control register (MDUC).

(1) Multiplication/division control register (MDUC)

MDUC is an 8-bit register that controls the operation of the multiplier/divider.

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

Reset signal generation clears this register to 00H.

Figure 16-5. Format of Multiplication/Division Control Register (MDUC)

Address: F00E8H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
MDUC	DIVMODE	0	0	0	0	0	0	DIVST

DIVMODE	Operation mode (multiplication/division) selection
0	Multiplication mode
1	Division mode

DIVST[Note]	Division operation start/stop
0	Division operation processing complete
1	Starts division operation/division operation processing in progress

Note DIVST can only be set (1) in the division mode. In the division mode, division operation is started by setting (1) DIVST. DIVST is automatically cleared (0) when the operation ends. In the multiplication mode, operation is automatically started by setting the multiplier and multiplicand to MDAH and MDAL, respectively.

Cautions 1. Do not rewrite DIVMODE during operation processing (while DIVST is 1). If it is rewritten, the operation result will be an undefined value.

2. DIVST cannot be cleared (0) by using software during division operation processing (while DIVST is 1).

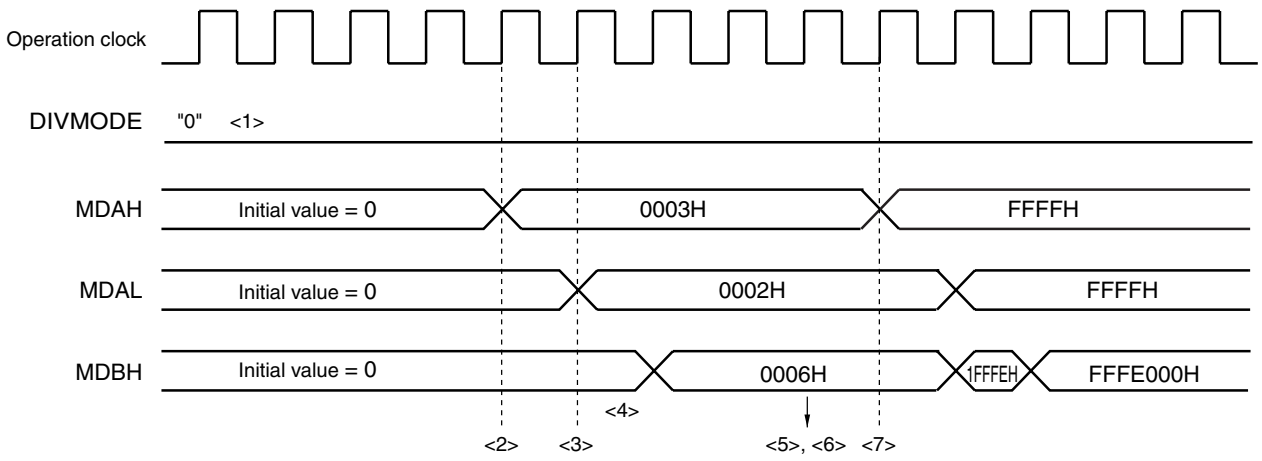
16.4 Operations of Multiplier/Divider

16.4.1 Multiplication operation

- Initial setting
  - <1> Set bit 7 (DIVMODE) of the multiplication/division control register (MDUC) to 0.
  - <2> Set the multiplicand to the multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to the 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 MDAH and MDAL, 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 the multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from the multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation
  - <7> To execute multiplication operation next, start from the “Initial setting” for multiplication operation.
  - <8> To execute division operation next, start from the “Initial setting” in **16.4.2 Division operation**.

Remark Steps <1> to <7> correspond to <1> to <7> in **Figure 16-6**.

Figure 16-6. Timing Diagram of Multiplication Operation (0003H × 0002H)



16.4.2 Division operation

- Initial setting
  - <1> Set bit 7 (DIVMODE) of the multiplication/division control register (MDUC) to 1.
  - <2> Set the dividend (higher 16 bits) to the multiplication/division data register A (H) (MDAH).
  - <3> Set the dividend (lower 16 bits) to the multiplication/division data register A (L) (MDAL).
  - <4> Set the divisor (higher 16 bits) to the multiplication/division data register B (H) (MDBH).
  - <5> Set the divisor (lower 16 bits) to the multiplication/division data register B (L) (MDBL).
  - <6> Set bit 0 (DIVST) of MDUC to 1.

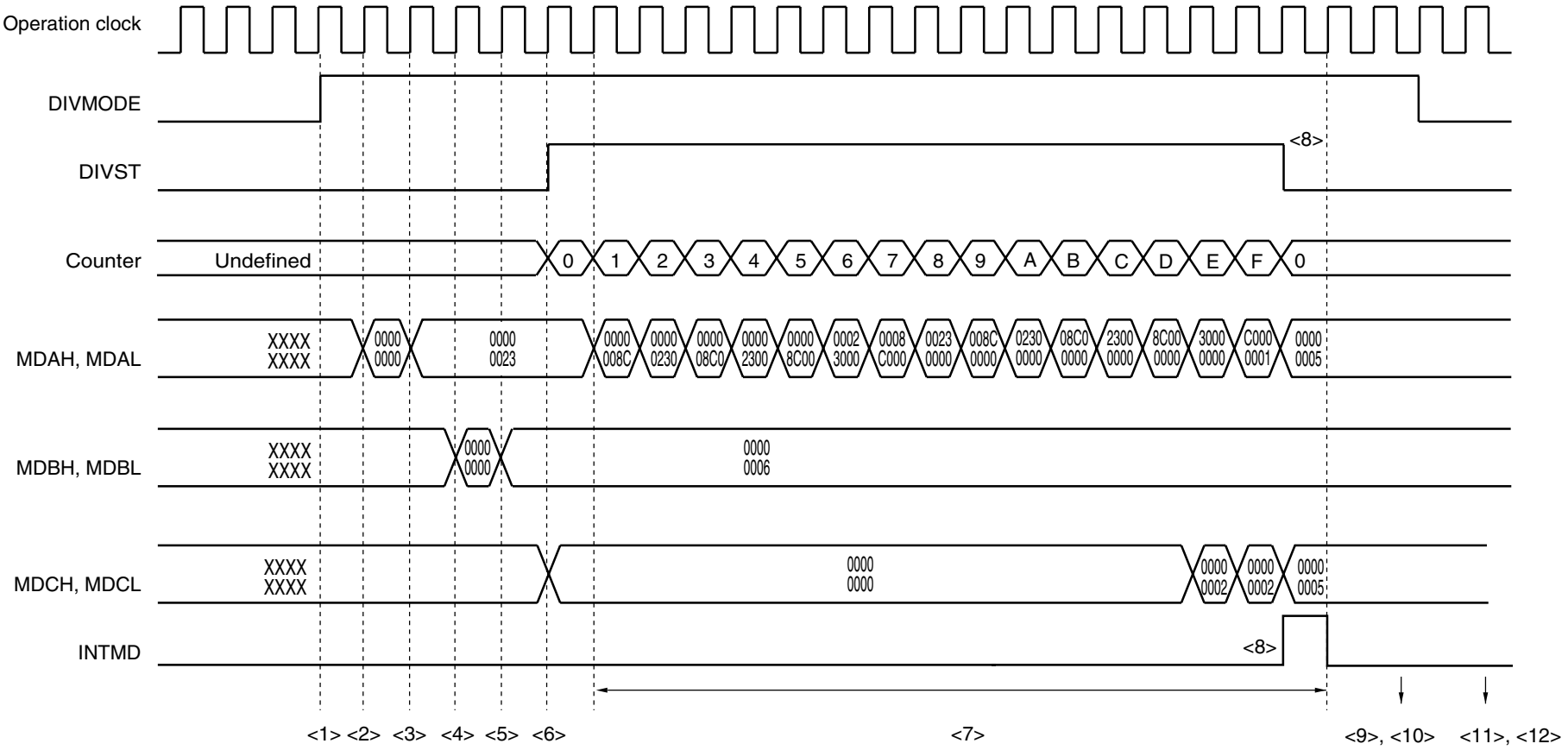
    (There is no preference in the order of executing steps <2> to <5>.)
- During operation processing
  - <7> The operation will end when one of the following processing is completed.
    - A wait of at least 16 clocks (The operation will end when 16 clocks have been issued.)
    - A check whether DIVST has been cleared
    - Generation of a division completion interrupt (INTMD)

    (The read values of MDBL, MDBH, MDCH, and MDCL during operation processing are not guaranteed.)
- Operation end
  - <8> DIVST is cleared (0) and an interrupt request signal (INTMD) is generated (end of operation).
  - <9> Read the quotient (lower 16 bits) from MDAL.
  - <10> Read the quotient (higher 16 bits) from MDAH.
  - <11> Read the remainder (lower 16 bits) from multiplication/division data register C (L) (MDCL).
  - <12> Read the remainder (higher 16 bits) from the multiplication/division data register C (H) (MDCH).

    (There is no preference in the order of executing steps <9> to <12>.)
- Next operation
  - <13> To execute multiplication operation next, start from the “Initial setting” in **16.4.1 Multiplication operation**.
  - <14> To execute division operation next, start from the “Initial setting” for division operation.

Remark Steps <1> to <12> correspond to <1> to <12> in **Figure 16-7**.

Figure 16-7. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



CHAPTER 17 DMA CONTROLLER

The DMA (Direct Memory Access) controller is mounted onto all 78K0R/Lx3-M microcontroller products.

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.

17.1 Functions of DMA Controller

- Number of DMA channels: 2
- 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 target: Between SFR and internal RAM
- Transfer request: Selectable from the following peripheral hardware interrupts
- Timer array unit (channels 0, 1, 4, 5)
- Serial array unit (UART0, UART3)
- Extended SFR (3rd SFR) interface (CSI10)
- 10-bit successive approximation type A/D converter

Here are examples of functions using DMA.

- Successive transfer of serial interface
- Batch transfer of analog data
- Capturing A/D conversion result at fixed interval
- Capturing port value at fixed interval

17.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

Table 17-1. Configuration of DMA Controller

Item	Configuration
Address registers	• DMA SFR address registers 0, 1 (DSA0, DSA1) • DMA RAM address registers 0, 1 (DRA0, DRA1)
Count register	• DMA byte count registers 0, 1 (DBC0, DBC1)
Control registers	• DMA mode control registers 0, 1 (DMC0, DMC1) • DMA operation control registers 0, 1 (DRC0, DRC1)

(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[Note].

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.

DSAn 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.

Note Except for address FFFFEH because the PMC register is allocated there.

Figure 17-1. Format of DMA SFR Address Register n (DSAn)

Address: FFFB0H (DSA0), FFFB1H (DSA1) After reset: 00H R/W

	7	6	5	4	3	2	1	0
DSAn								

Remark n: DMA channel number (n = 0, 1)

(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 (FE300H to FFEDFH) can be set to this register.

Set the lower 16 bits of the RAM address.

This register is automatically incremented when DMA transfer has been started. It is incremented by +1 in the 8-bit transfer mode and by +2 in the 16-bit transfer mode. DMA transfer is started from the address set to this DRAn register. When the data of the last address has been transferred, DRAn 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.

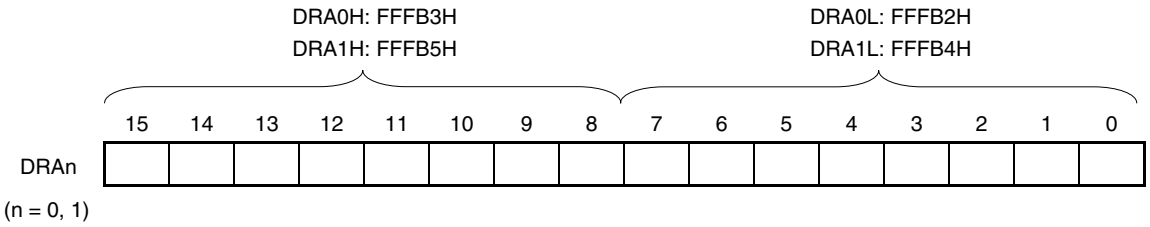
In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

DRAn 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 17-2. Format of DMA RAM Address Register n (DRAn)

Address: FFFB2H, FFFB3H (DRA0), FFFB4H, FFFB5H (DRA1) After reset: 0000H R/W



Remark n: DMA channel number (n = 0, 1)

(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.

DBCn 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 17-3. Format of DMA Byte Count Register n (DBCn)

Address: FFFB6H, FFFB7H (DBC0), FFFB8H, FFFB9H (DBC1) After reset: 0000H R/W

DBC0H: FFFB7H
DBC1H: FFFB9H

DBC0L: FFFB6H
DBC1L: FFFB8H

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

(n = 0, 1)

DBCn[9:0]	Number of Times of Transfer (When DBCn is Written)	Remaining Number of Times of Transfer (When DBCn is Read)
000H	1024	Completion of transfer or waiting for 1024 times of DMA transfer
001H	1	Waiting for remaining one time of DMA transfer
002H	2	Waiting for remaining two times of DMA transfer
003H	3	Waiting for remaining three times of DMA transfer
• • •	• • •	• • •
3FEH	1022	Waiting for remaining 1022 times of DMA transfer
3FFH	1023	Waiting for remaining 1023 times of DMA transfer

Cautions 1. Be sure to clear bits 15 to 10 to "0".

2. If the general-purpose register is specified or the internal RAM space is exceeded as a result of continuous transfer, the general-purpose register or SFR space are written or read, resulting in loss of data in these spaces. Be sure to set the number of times of transfer that is within the internal RAM space.

Remark n: DMA channel number (n = 0, 1)

17.3 Registers Controlling DMA Controller

DMA controller is controlled by the following registers.

- DMA mode control register n (DMCn)
- DMA operation control register n (DRCn)

Remark n: DMA channel number (n = 0, 1)

(1) DMA mode control register n (DMCn)

DMCn 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 DMCn is prohibited during operation (when DSTn = 1).
DMCn can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 17-4. Format of DMA Mode Control Register n (DMCn) (1/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

STGn[Note 1]	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).
DMA transfer is started by writing 1 to STGn when DMA operation is enabled (DENn = 1). When this bit is read, 0 is always read.	

DRSn	Selection of DMA transfer direction
0	SFR to internal RAM
1	Internal RAM to SFR

DSn	Specification of transfer data size for DMA transfer
0	8 bits
1	16 bits

DWAITn[Note 2]	Pending of DMA transfer
0	Executes DMA transfer upon DMA start request (not held pending).
1	Holds DMA start request pending if any.
DMA transfer that has been held pending can be started by clearing the value of DWAITn to 0. It takes 2 clocks to actually hold DMA transfer pending when the value of DWAITn is set to 1.	

Notes 1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

2. When DMA transfer is held pending while using both DMA channels, be sure to hold the DMA transfer pending for both channels (by setting the DWAIT0 and DWAIT1 bits to 1).

Remark n: DMA channel number (n = 0, 1)

Figure 17-4. Format of DMA Mode Control Register n (DMCn) (2/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

IFCn 3	IFCn 2	IFCn 1	IFCn 0	Selection of DMA start source[Note]	
				Trigger signal	Trigger contents
0	0	0	0	–	Disables DMA transfer by interrupt. (Only software trigger is enabled.)
0	0	1	0	INTTM00	Timer channel 0 interrupt
0	0	1	1	INTTM01	Timer channel 1 interrupt
0	1	0	0	INTTM04	Timer channel 4 interrupt
0	1	0	1	INTTM05	Timer channel 5 interrupt
0	1	1	0	INTST0	UART0 transmission end interrupt
0	1	1	1	INTSR0	UART0 reception end interrupt
1	0	0	0	INTCSI10	CSI10 transfer end interrupt
1	0	1	0	INTST3	UART3 transmission end interrupt
1	0	1	1	INTSR3	UART3 reception end interrupt
1	1	0	0	INTAD	A/D conversion end interrupt
Other than above				Setting prohibited	

Note The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 values.

Remark n: DMA channel number (n = 0, 1)

(2) DMA operation control register n (DRCn)

DRCn 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).

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

Reset signal generation clears this register to 00H.

Figure 17-5. Format of DMA Operation Control Register n (DRCn)

Address: FFFBCH (DRC0), FFFBDH (DRC1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA operation enable flag
0	Disables operation of DMA channel n (stops operating cock of DMA).
1	Enables operation of DMA channel n.
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1).	

DSTn	DMA transfer mode flag
0	DMA transfer of DMA channel n is completed.
1	DMA transfer of DMA channel n is not completed (still under execution).
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1). When a software trigger (STGn) or the start source trigger set by IFCn3 to IFCn0 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.	

Cautions 1. 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 DSTn to 0 and then DENn to 0 (for details, refer to 17.5.6 Forced termination by software).

2. When the FSEL bit of the OSMC register has been set to 1, do not enable (DENn = 1) DMA operation for at least three clocks after the setting.

Remark n: DMA channel number (n = 0, 1)

17.4 Operation of DMA Controller

17.4.1 Operation procedure

<1> The DMA controller is enabled to operate when DENn = 1. Before writing the other registers, be sure to set DENn 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 the DSAn, DRAn, DBCn, and DMCn registers.

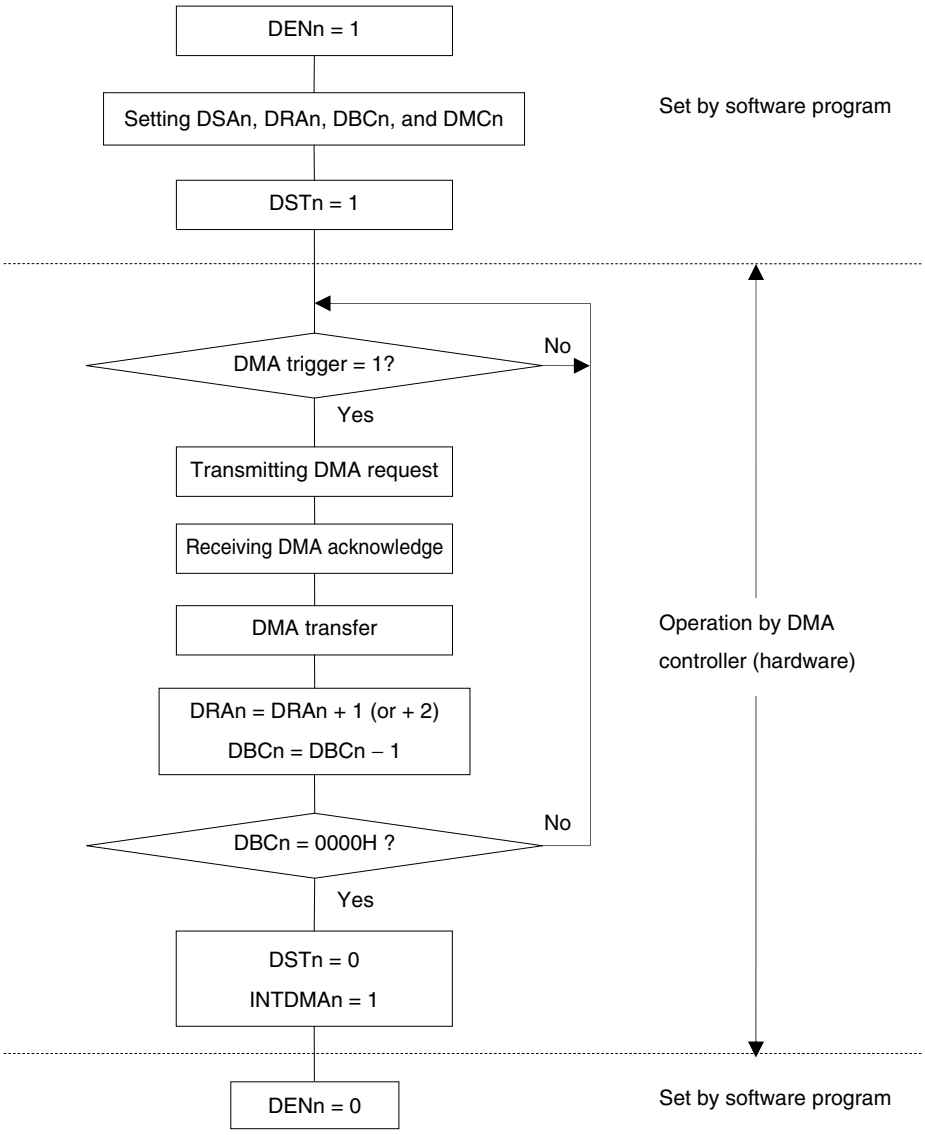
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.

<4> When a software trigger (STGn) or a start source trigger specified by IFCn3 to IFCn0 is input, a DMA transfer is started.

<5> Transfer is completed when the number of times of transfer set by the DBCn register reaches 0, and transfer is automatically terminated by occurrence of an interrupt (INTDMAn).

<6> Stop the operation of the DMA controller by clearing DENn to 0 when the DMA controller is not used.

Figure 17-6. Operation Procedure



Remark n: DMA channel number (n = 0, 1)

17.4.2 Transfer mode

The following four modes can be selected for DMA transfer by using bits 6 and 5 (DRSn and DSn) of the DMCn register.

DRSn	DSn	DMA Transfer Mode
0	0	Transfer from SFR of 1-byte data (fixed address) to RAM (address is incremented by +1)
0	1	Transfer from SFR of 2-byte data (fixed address) to RAM (address is incremented by +2)
1	0	Transfer from RAM of 1-byte data (address is incremented by +1) to SFR (fixed address)
1	1	Transfer from RAM of 2-byte data (address is incremented by +2) to SFR (fixed address)

By using these transfer modes, up to 1024 bytes of data can be consecutively transferred by using the serial interface, data resulting from A/D conversion can be consecutively transferred, and port data can be scanned at fixed time intervals by using a timer.

17.4.3 Termination of DMA transfer

When DBCn = 00H and DMA transfer is completed, the DSTn bit is automatically cleared to 0. An interrupt request (INTDMAn) is generated and transfer is terminated.

When the DSTn bit is cleared to 0 to forcibly terminate DMA transfer, the DBCn and DRAn registers hold the value when transfer is terminated.

The interrupt request (INTDMAn) is not generated if transfer is forcibly terminated.

Remark n: DMA channel number (n = 0, 1)

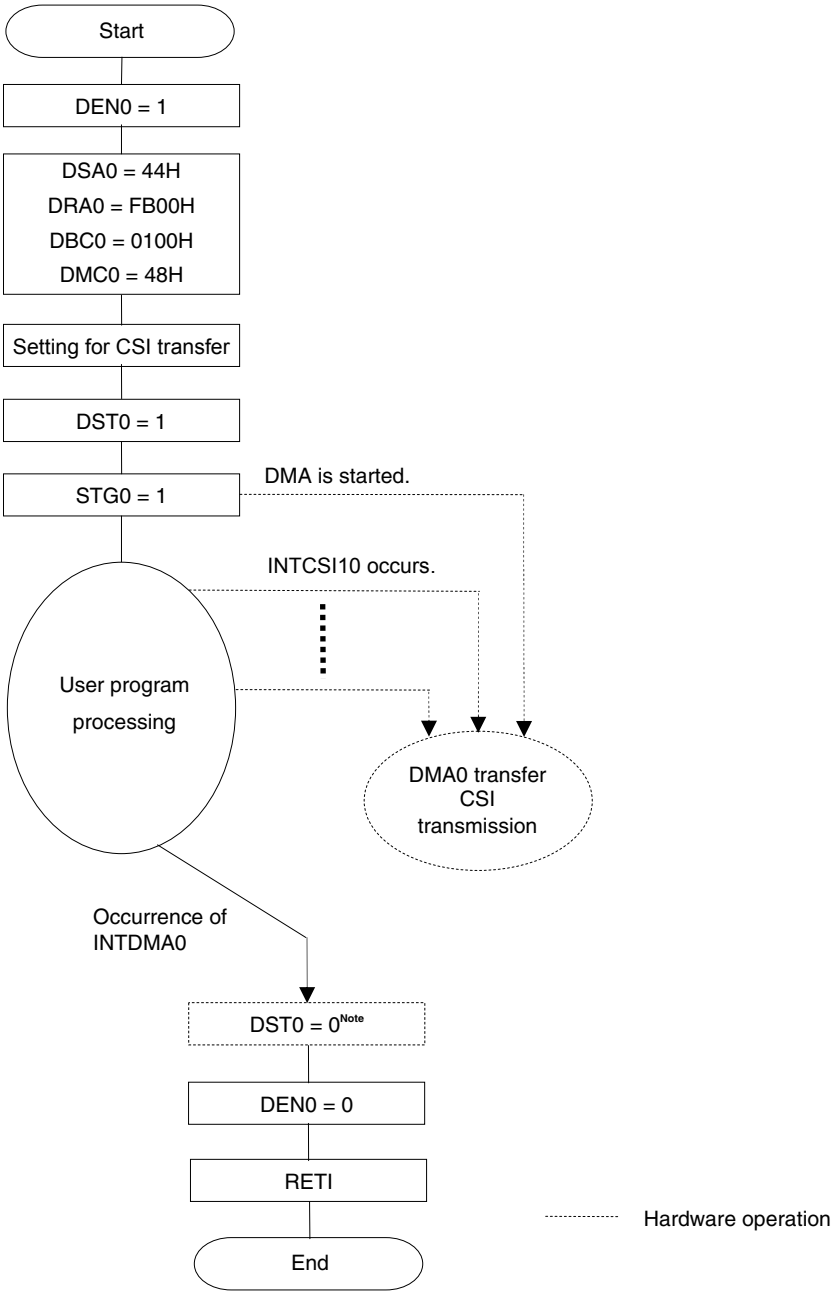
17.5 Example of Setting of DMA Controller

17.5.1 Writing to extended SFR (3rd SFR) consecutive address

A flowchart showing an example of setting for writing to the extended SFR (3rd SFR) consecutive address is shown below.

- Consecutive transmission of CSI10
- 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 (bits 3 to 0 of the DMC0 register) = 1000B.
- Transfers FFB00H to FFBFFH (256 bytes) of RAM to FFF44H of the transmit buffer (SIO10) of CSI.

Figure 17-7. Example of Setting for Writing to Extended SFR (3rd SFR) Consecutive Address



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 DST0 to 0 and then DEN0 to 0 (for details, refer to **17.5.6 Forced termination by software**).

The first trigger for consecutive transmission is not started by the interrupt of CSI. In this example, it start by a software trigger.

CSI transmission of the second time and onward is automatically executed.

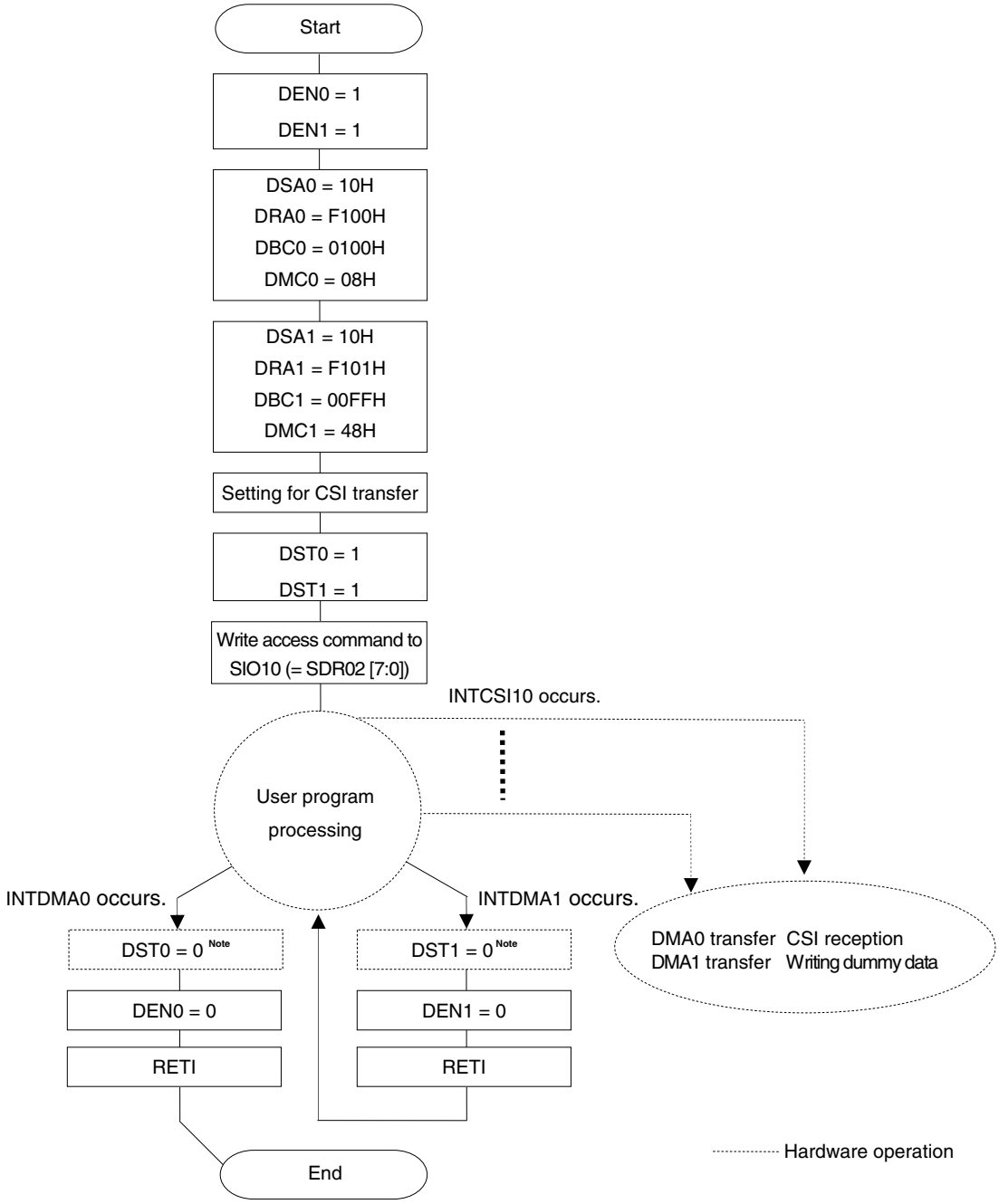
A DMA interrupt (INTDMA0) occurs when the last transmit data has been written to the data register.

17.5.2 Reading to extended SFR (3rd SFR) consecutive address

A flowchart showing an example of setting for reading to the extended SFR (3rd SFR) consecutive address is shown below.

- Master reception of CSI10
- DMA channel 0 is used to read received data and DMA channel 1 is used to write dummy data.
- DMA start source: INTCSI10
  (If the same start source is specified for DMA channels 0 and 1, the data of channel 0 is transferred, and then that of channel 1.)
- Interrupt of CSI10 is specified by IFC03 to IFC00 = IFC13 to IFC10 (bits 3 to 0 of the DMCn register) = 1000B.
- The access command is written to the first byte by using software (an instruction).
- Data is transferred (received) from FFF10H of the CSI data register (SIO10) to FF100H to FF1FFH of RAM (256 bytes). (In successive reception mode, the data that is to be received when the first buffer empty interrupt occurs is invalid because the data has not been received.)
- Transfers dummy data FF101H to FF1FFH (255 bytes) of RAM to FFF10H of the data register (SIO10) of CSI.

Figure 17-8. Example of Setting for Reading to Extended SFR (3rd SFR) Consecutive Address



Note The DSTn flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DENn flag is enabled only when DSTn = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMAn (INTDMAn), set DSTn to 0 and then DENn to 0 (for details, refer to **17.5.6 Forcible termination by software**).

The received data is automatically transferred from the first byte. In successive reception mode, the data that is to be received when the first buffer empty interrupt occurs is invalid because the valid data has not been received.

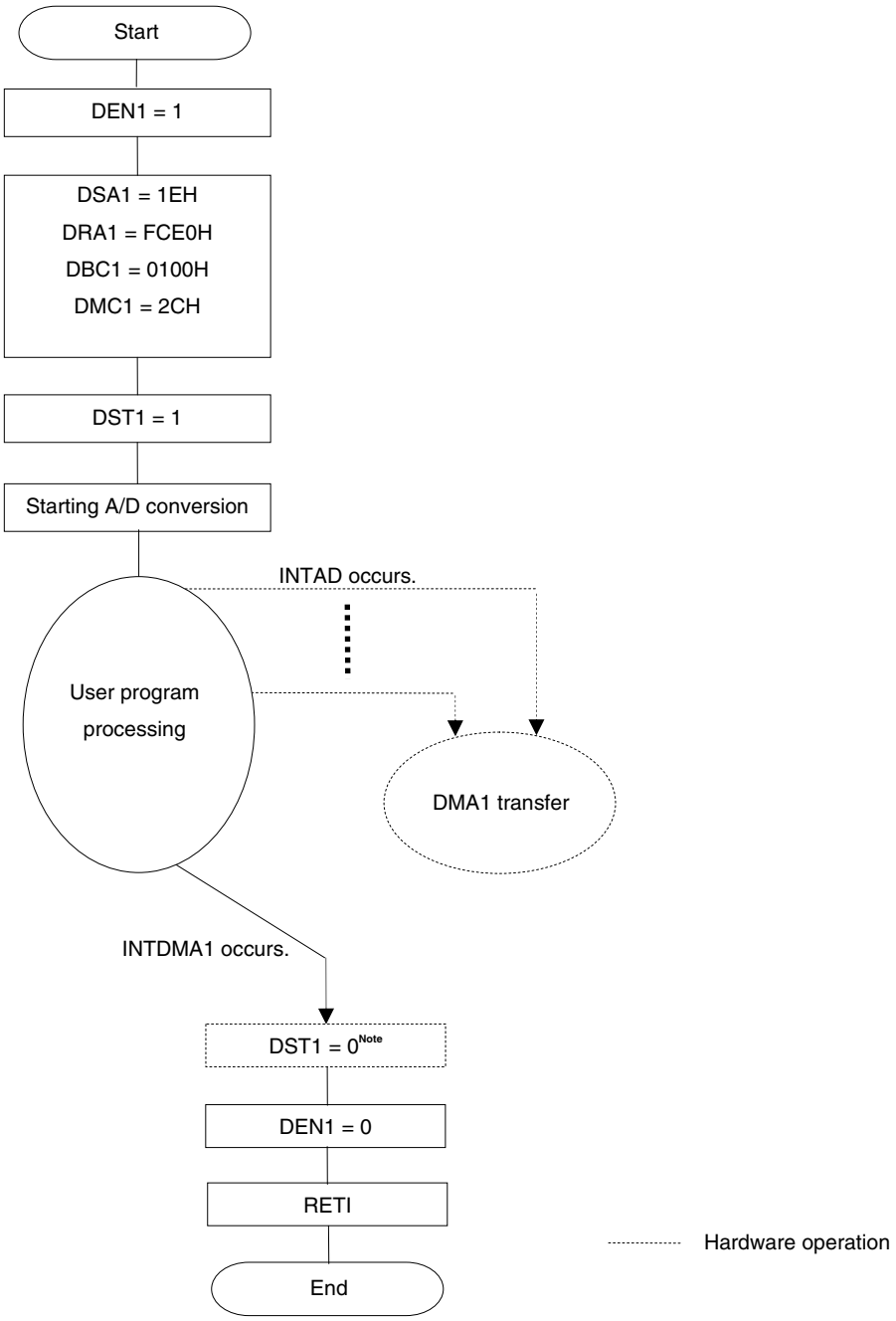
A DMA interrupt (INTDMA1) occurs when the last dummy data has been writing to the data register. A DMA interrupt (INTDMA0) occurs when the last received data has been read from the data register. To restart the DMA transfer, the CSI transfer must be completed.

17.5.3 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 (bits 3 to 0 of the DMC1 register) = 1100B.
- Transfers FFF1EH and FFF1FH (2 bytes) of the 10-bit A/D conversion result register to 512 bytes of FFCE0H to FFEDFH of RAM.

Figure 17-9. Example of Setting of Consecutively Capturing A/D Conversion Results



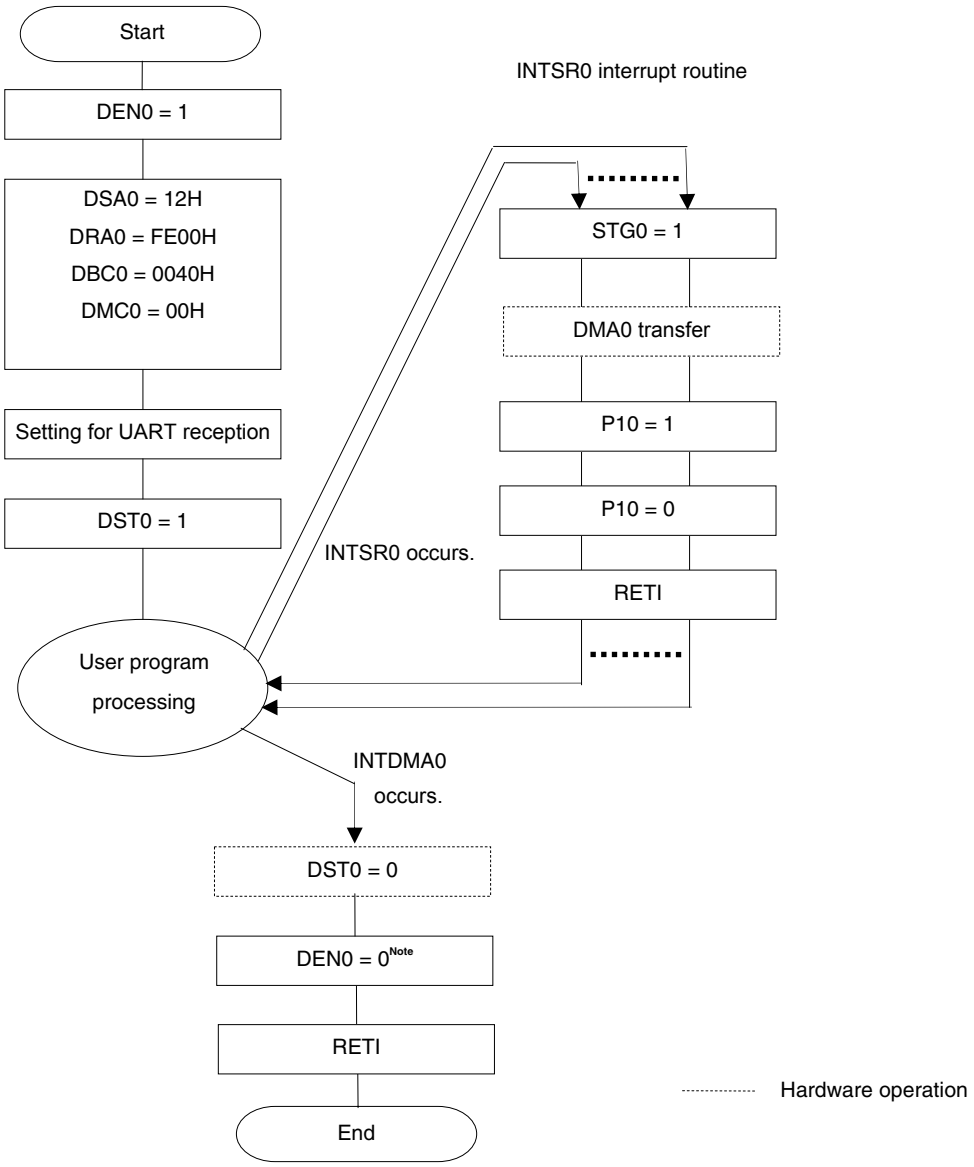
Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set DST1 to 0 and then DEN1 to 0 (for details, refer to **17.5.6 Forced termination by software**).

17.5.4 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 17-10. 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 DST0 to 0 and then DEN0 to 0 (for details, refer to **17.5.6 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.

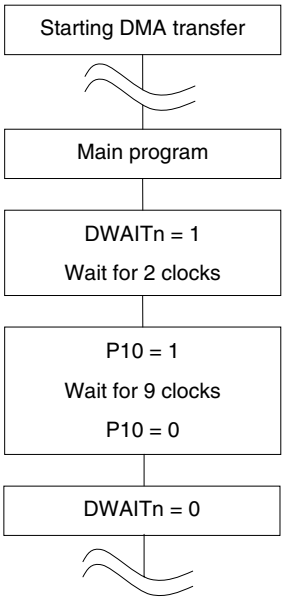
17.5.5 Holding DMA transfer pending by DWAITn

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 17-11. Example of Setting for Holding DMA Transfer Pending by DWAITn



Caution When DMA transfer is held pending while using both DMA channels, be sure to held the DMA transfer pending for both channels (by setting DWAIT0 and DWAIT1 to 1). If the DMA transfer of one channel is executed while that of the other channel is held pending, DMA transfer might not be held pending for the latter channel.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

17.5.6 Forced termination by software

After DSTn is set to 0 by software, it takes up to 2 clocks until a DMA transfer is actually stopped and DSTn is set to 0. To forcibly terminate a DMA transfer by software without waiting for occurrence of the interrupt (INTDMAn) of DMAn, therefore, perform either of the following processes.

<When using one DMA channel>

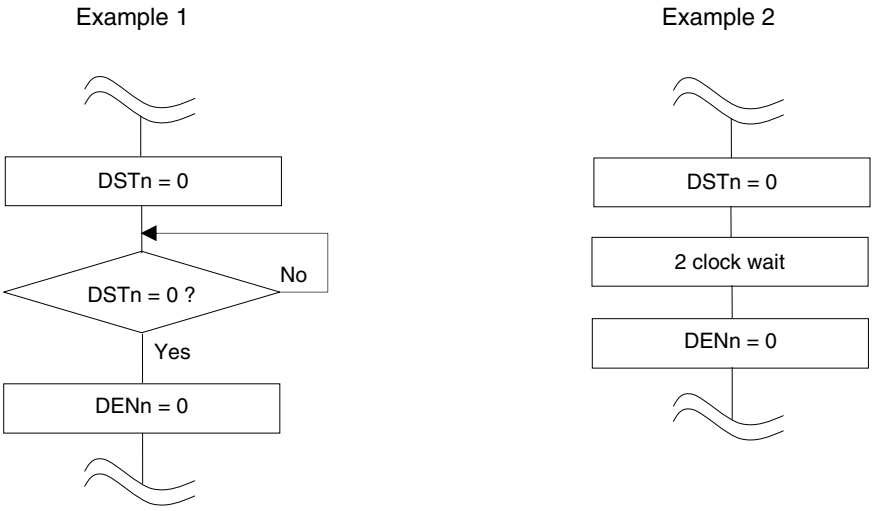
- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software, confirm by polling that the DSTn bit has actually been cleared to 0, and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction).

- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction) two or more clocks after.

<When using both DMA channels>

- To forcibly terminate DMA transfer by software when using both DMA channels (by setting DSTn to 0), clear the DSTn bit to 0 after the DMA transfer is held pending by setting the DWAIT0 and DWAIT1 bits of both channels to 1. Next, clear the DWAIT0 and DWAIT1 bits of both channels to 0 to cancel the pending status, and then clear the DENn bit to 0.

Figure 17-12. Forced Termination of DMA Transfer (1/2)



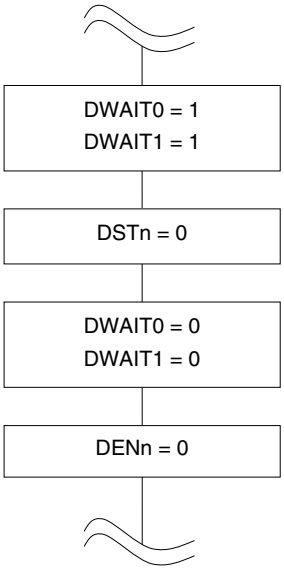
Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

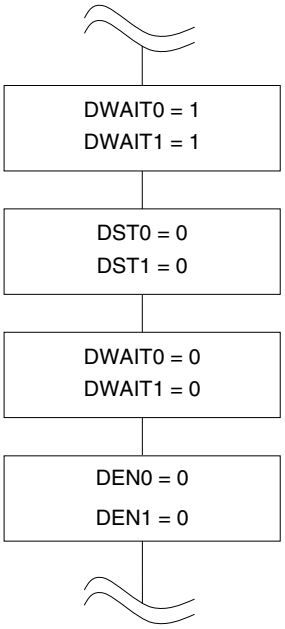
Figure 17-12. Forced Termination of DMA Transfer (2/2)

Example 3

- Procedure for forcibly terminating the DMA transfer for one channel if both channels are used



- Procedure for forcibly terminating the DMA transfer for both channels if both channels are used



Caution In example 3, the system is not required to wait two clock cycles after the DWAITn bit is set to 1. In addition, the system does not have to wait two clock cycles after clearing the DSTn bit to 0, because more than two clock cycles elapse from when the DSTn bit is cleared to 0 to when the DENn bit is cleared to 0.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

17.6 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 DMA requests are generated at the same time, however, DMA channel 0 takes priority over DMA channel 1.

If a DMA request and an interrupt request are generated at the same time, the DMA transfer takes precedence, and then interrupt servicing is executed.

(2) DMA response time

The response time of DMA transfer is as follows.

Table 17-2. Response Time of DMA Transfer

	Minimum Time	Maximum Time
Response time	3 clocks	10 clocks[Note]

Note The maximum time necessary to execute an instruction from internal RAM is 16 clock cycles.

Cautions 1. The above response time does not include the two clock cycles required for a DMA transfer.

2. When executing a DMA pending instruction (see 17.6 (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 17-3. DMA Operation in Standby Mode

Status	DMA Operation
HALT mode	Normal operation
STOP mode	Stops operation. If DMA transfer and STOP instruction execution contend, DMA transfer may be damaged. Therefore, stop DMA before executing the STOP instruction.

(4) DMA pending instruction

Even if a DMA request is generated, DMA transfer is held pending immediately after the following instructions.

- CALL !addr16
- CALL $!addr20
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- Bit manipulation instructions for registers IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H and PSW each.

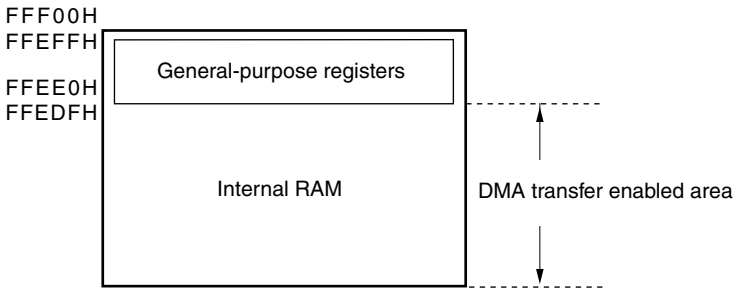
(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified

The address indicated by DRA0n 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.



CHAPTER 18 INTERRUPT FUNCTIONS

18.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the priority of vectored interrupt servicing. For the priority order, see **Table 18-1**.

A standby release signal is generated and STOP and HALT modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

18.2 Interrupt Sources and Configuration

The interrupt sources consist of maskable interrupts and software interrupts. In addition, they also have up to five reset sources (see **Table 18-1**). 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 18-1. Interrupt Source List (1/2)

Interrupt Type	Internal/ External	Basic Configuration Type[Note 1]	Default Priority[Note 2]	Interrupt Source		Vector Table Address
				Name	Trigger	
Maskable	Internal	(A)	0	INTWDTI	Watchdog timer interval[Note 3] (75% of overflow time)	00004H
			1	INTLVI	Low-voltage detection[Note 4]	00006H
	External	(B)	2	INTP0	Pin input edge detection	00008H
			3	INTP3	Pin input edge detection	0000EH
	Internal	(A)	4	INTST3	End of UART3 transmission	00014H
			5	INTSR3	End of UART3 reception	00016H
			6	INTSRE3	UART3 communication error occurrence	00018H
			7	INTDMA0	End of DMA0 transfer	0001AH
			8	INTDMA1	End of DMA1 transfer	0001CH
			9	INTST0	End of UART0 transmission	0001EH
			10	INTSR0	End of UART0 reception	00020H
			11	INTSRE0	UART0 communication error occurrence	00022H
			12	INTCSI10	End of CSI10 communication	00024H
			13	INTIICA	End of IICA communication	0002AH
			14	INTTM00	End of timer channel 0 count or capture	0002CH
			15	INTTM01	End of timer channel 1 count or capture	0002EH
			16	INTTM02	End of timer channel 2 count or capture	00030H
			17	INTTM03	End of timer channel 3 count or capture	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 35 indicates the lowest priority.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 18-1.

3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

4. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is cleared to 0.

Table 18-1. Interrupt Source List (2/2)

Interrupt Type	Internal/ External	Basic Configuration Type[Note 1]	Default Priority[Note 2]	Interrupt Source		Vector Table Address
				Name	Trigger	
Maskable	Internal	(A)	18	INTAD	End of A/D conversion	00034H
			19	INTRTC	Fixed-cycle signal of real-time counter 2/alarm match detection	00036H
			20	INTRTCI	Interval signal detection of real-time counter 2	00038H
			21	INTST2	End of UART2 transmission/	0003CH
				INTCSI20	End of CSI20 communication/	
				INTIIC20	End of IIC20 communication	
			22	INTSR2	End of UART2 reception	0003EH
			23	INTSRE2	UART2 communication error occurrence	00040H
			24	INTTM04	End of timer channel 4 count or capture	00042H
			25	INTTM05	End of timer channel 5 count or capture	00044H
			26	INTTM06	End of timer channel 6 count or capture	00046H
			27	INTTM07	End of timer channel 7 count or capture	00048H
	External	(B)	28	INTP6	Pin input edge detection	0004AH
			29	INTP8	Interrupt request from extended SFR (3rd SFR)	0004EH
			30	INTP9	Pin input edge detection	00050H
	Internal	(A)	31	INTTM10	End of timer channel 10 count or capture	00056H
			32	INTTM11	End of timer channel 11 count or capture	00058H
			33	INTTM12	End of timer channel 12 count or capture	0005AH
			34	INTTM13	End of timer channel 13 count or capture	0005CH
			35	INTMD	End of multiply/divide operation	0005EH
Software	–	(C)	–	BRK	Execution of BRK instruction	0007EH
Reset	–	–	–	RESET	$\overline{\text{RESET}}$ pin input	00000H
				POC	Power-on-clear	
				LVI	Low-voltage detection[Note 3]	
				WDT	Overflow of watchdog timer	
				TRAP	Execution of illegal instruction[Note 4]	

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 35 indicates the lowest priority.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 18-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

4. 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.

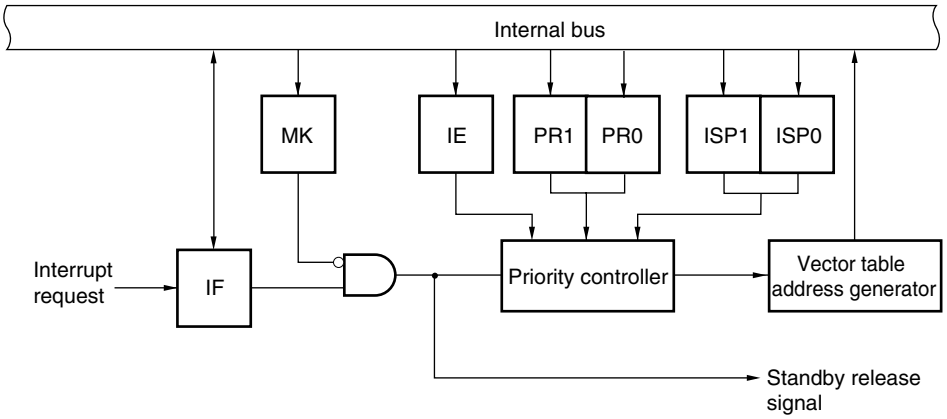
If an interrupt is generated by any circuit controlled using an extended SFR (3rd SFR) (the real-time counter, ΔΣ-type A/D converter, power calculation circuit, power quality measurement circuit, or digital frequency conversion circuit), the interrupt is input to the interrupt controller as an INTP8 interrupt. Check the interrupt source in the interrupt handler, and then perform the appropriate processing.

Table 18-2. Interrupt Source List of Extended SFR (3rd SFR)

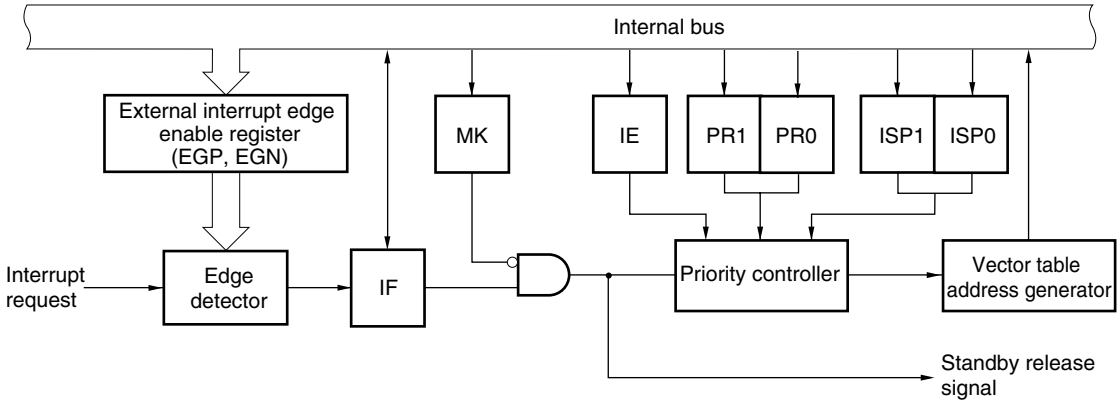
Interrupt Type	Internal/ External	Interrupt Source		Circuit
		Name	Trigger	
Maskable	Internal (extended SFR (3rd SFR))	INTRTC0	RTC alarm and constant-period interrupt	Real-time counter
		INTRTC2	RTC interval interrupt	Real-time counter
		INTAD2	ΔΣ ADC conversion end interrupt	ΔΣ-type A/D converter
		INTFAULTSIGN	Fault detection interrupt	Power quality measurement circuit
		INTREASIGN	Reactive power sign change interrupt	Power calculation circuit
		INTACTSIGN	Active power sign change interrupt	Power calculation circuit
		INTAPPNOLD	Apparent power no-load interrupt	Power calculation circuit
		INTREANOLD	Reactive power no-load interrupt	Power calculation circuit
		INTACTNOLD	Active power no-load interrupt	Power calculation circuit
		INTCF	CF output interrupt	Digital frequency conversion circuit
		INTAPPEOF	APPHR register overflow interrupt	Power calculation circuit
		INTREAEOF	REAHR register overflow interrupt	Power calculation circuit
		INTACTEOF	ACTHR register overflow interrupt	Power calculation circuit
		INTAPPEHF	APPHR register half full interrupt	Power calculation circuit
		INTREAEHF	REAHR register half full interrupt	Power calculation circuit
		INTACTEHF	ACTHR register half full interrupt	Power calculation circuit
		INTWFSM	WAVE form register update interrupt	Power calculation circuit
		INTPKI1	Peak detection interrupt for current channel	Power quality measurement circuit
		INTPKV1	Peak detection interrupt for voltage channel	Power quality measurement circuit
		INTCYCEND	Synchronous reading interrupt (accumulation completion interrupt of LINNUM)	Power calculation circuit
		INTZXTO1	Zero-crossing timeout interrupt (voltage channel 1)	Power quality measurement circuit
		INTZXTO2	Zero-crossing timeout interrupt (voltage channel 2)	Power quality measurement circuit
		INTZX1	Zero-crossing detection interrupt (voltage channel 1)	Power quality measurement circuit
		INTZX2	Zero-crossing detection interrupt (voltage channel 2)	Power quality measurement circuit
		INTSAG1	SAG detection interrupt (voltage channel 1)	Power quality measurement circuit
		INTSAG2	SAG detection interrupt (voltage channel 2)	Power quality measurement circuit
		INTRQFLG	Interrupt from extended SFR (3rd SFR) space	–

Figure 18-1. Basic Configuration of Interrupt Function

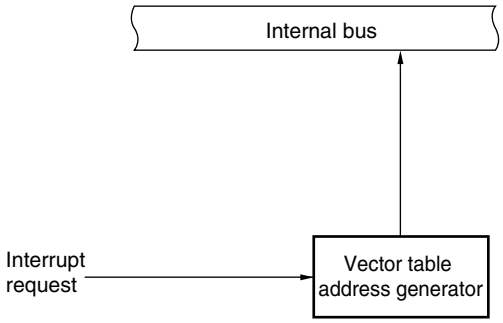
(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)



(C) 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

18.3 Registers Controlling Interrupt Functions

The following 8 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- Extended SFR (3rd SFR) interrupt request flag registers (IF20, IF21, IF22, IF23)
- Extended SFR (3rd SFR) Interrupt mask flag registers (MK20, MK21, MK22, MK23)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)
- External interrupt rising edge enable registers (EGP0, EGP1)
- External interrupt falling edge enable registers (EGN0, EGN1)
- Program status word (PSW)

Table 18-3 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 18-3. Flags Corresponding to Interrupt Request Sources (1/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1	
INTP0	PIF0		PMK0		PPR00, PPR10	
INTP3	PIF3		PMK3		PPR03, PPR13	
INTST3	STIF3	IF0H	STMK3	MK0H	STPR03, STPR13	PR00H, PR10H
INTSR3	SRIF3		SRMK3		SRPR03, SRPR13	
INTSRE3	SREIF3		SREMK3		SREPR03, SREPR13	
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10	
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11	
INTST0	STIF0		STMK0		STPR00, STPR10	
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10	
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10	
INTCSI10	CSIIF10	IF1L	CSIMK10	MK1L	CSIPR010, CSIPR110	PR01L, PR11L
INTIICA	IICAIF		IICAMK		IICAPR0, IICAPR1	
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102	
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103	

Table 18-3. Flags Corresponding to Interrupt Request Sources (2/2)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1	
INTRTCI	RTCIIF		RTCIMK		RTCIPR0, RTCIPR1	
INTST2[Note]	STIF2[Note]		STMK2[Note]		STPR02, STPR12[Note]	
INTCSI20[Note]	CSIIF20[Note]		CSIMK20[Note]		CSIPR020, CSIPR120[Note]	
INTIIC20[Note]	IICIF20[Note]		IICMK20[Note]		IICPR020, IICPR120[Note]	
INTSR2	SRIF2		SRMK2		SRPR02, SRPR12	
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12	
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104	
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106	
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107	
INTP6	PIF6		PMK6		PPR06, PPR16	
INTP8	PIF8		PMK8		PPR08, PPR18	
INTP9	PIF9		PMK9		PPR09, PPR19	
INTTM10	TMIF10	IF2H	TMMK10	MK2H	TMPR010, TMPR110	PR02H, PR12H
INTTM11	TMIF11		TMMK11		TMPR011, TMPR111	
INTTM12	TMIF12		TMMK12		TMPR012, TMPR112	
INTTM13	TMIF13		TMMK13		TMPR013, TMPR113	
INTMD	MDIF		MDMK		MDPR0, MDPR1	

Note Do not use UART2, CSI20, and IIC20 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST2, INTCSI20, and INTIIC20 is generated, bit 4 of IF1H is set to 1. Bit 4 of MK1H, PR01H, and PR11H supports these three interrupt sources.

Table 18-4. Flags Corresponding to Extended SFR (3rd SFR) Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag	
		Register		Register
INTRTC0	RTC0IF	IF20	RTC0MK	MK20
INTRTC2	RTC2IF		RTC2MK	
INTAD2	ADIF2		ADMK2	
INTFAULTSIGN	FAULTSIGNIF		FAULTSIGNMK	
INTREASIGN	REASIGNIF		REASIGNMK	
INTACTSIGN	ACTSIGNIF		ACTSIGNMK	
INTAPPNOLD	APPNOLDIF		APPNOLDMK	
INTREANOLD	REANOLDIF		REANOLDMK	
INTACTNOLD	ACTNOLDIF	IF21	ACTNOLDMK	MK21
INTCF	CFIF		CFMK	
INTAPPEOF	APPEOFIF		APPEOFMK	
INTREAEOF	REAEOFIF		REAEOFMK	
INTACTEOF	ACTEOFIF		ACTEOFMK	
INTAPPEHF	APPEHFIF		APPEHFMK	
INTREAEHF	REAEHFIF		REAEHFMK	
INTACTEHF	ACTEHFIF	IF22	ACTEHFMK	MK22
INTWFSM	WFSMIF		WFSMMK	
INTPKI1	PKI1IF		PKI1MK	
INTPKV1	PKV1IF		PKV1MK	
INTCYCEND	CYCENDIF		CYCENDMK	
INTZXTO1	ZXTO1IF		ZXTO1MK	
INTZXTO2	ZXTO2IF	IF23	ZXTO2MK	MK23
INTZX1	ZX1IF		ZX1MK	
INTZX2	ZX2IF		ZX2MK	
INTSAG1	SAG1IF		SAG1MK	
INTSAG2	SAG2IF		SAG2MK	
INTRQFLG	–		–	

(1) Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

IF0L, IF0H, IF1L, IF1H, IF2L, and IF2H can be set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H, IF1L and IF1H, and IF2L and IF2H 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.

Cautions 1. When operating a timer, serial interface, or 10-bit successive approximation type A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

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 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.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 18-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

Address: FFFE0H After reset: 00H R/W

Symbol	7	6	<5>	4	3	<2>	<1>	<0>
IF0L	0	0	PIF3	0	0	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0	SRIF0	STIF0	DMAIF1	DMAIF0	SREIF3	SRIF3	STIF3

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	IICAIF	0	0	CSIIF10

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
IF1H	TMIF04	SREIF2	SRIF2	CSIIF20 IICIF20 STIF2	0	RTCIIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	<1>	<0>
IF2L	0	PIF9	PIF8	0	PIF6	TMIF07	TMIF06	TMIF05

Address: FFFD1H After reset: 00H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	0
IF2H	0	0	MDIF	TMIF13	TMIF12	TMIF11	TMIF10	0

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution Be sure to clear bits 3, 4, 6, 7 of IF0L, bits 1, 2 of IF1L, bit 3 of IF1H, bits 4, 7 of IF2L, bits 0, 6, 7 of IF2H to 0.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H can be set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, MK1L and MK1H, and MK2L and MK2H 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 18-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

Address: FFFE4H After reset: FFH R/W

Symbol	7	6	<5>	4	3	<2>	<1>	<0>
MK0L	1	1	PMK3	1	1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0	SRMK0	STMK0	DMAMK1	DMAMK0	SREMK3	SRMK3	STMK3

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	IICAMK	1	1	CSIMK10

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
MK1H	TMMK04	SREMK2	SRMK2	CSIMK20 IICMK20 STMK2	1	RTCIMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	7	<6>	<5>	4	<3>	<2>	<1>	<0>
MK2L	1	PMK9	PMK8	1	PMK6	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	0
MK2H	1	1	MDMK	TMMK13	TMMK12	TMMK11	TMMK10	1

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bits 3, 4, 6, 7 of MK0L, bits 1, 2 of MK1L, bit 3 of MK1H, bits 4, 7 of MK2L, bits 0, 6, 7 of MK2H to 1.

(3) Extended SFR (3rd SFR) interrupt request flag registers (IF20, IF21, IF22, IF23)

The extended SFR (3rd SFR) interrupt request flags are set to 1 when the corresponding interrupt request is generated. 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 not automatically cleared.

IF20, IF21, IF22, and IF23 are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears these registers to 00H.

Caution If clearing the extended SFR (3rd SFR) interrupt request flag before the extended SFR (3rd SFR) interrupt mask flag, clear the extended SFR (3rd SFR) interrupt request flag twice.

Figure 18-4. Format of Extended SFR (3rd SFR) Interrupt Request Flag Registers (IF20, IF21, IF22, IF23)

Address: 80H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF20	REANOLDIF	APPNOLDIF	ACTSIGNIF	REASIGNIF	FAULTSIGNIF	ADIF2	RTC2IF	RTC0IF

Address: 81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF21	REAEHFIF	APPEHFIF	ACTEOFIF	REAEOFIF	APPEOFIF	0	CFIF	ACTNOLDIF

Address: 82H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF22	ZXTO1IF	CYCENDIF	0	PKV1IF	0	PKI1IF	WFSMIF	ACTEHFIF

Address: 83H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF23	0	0	0	SAG2IF	SAG1IF	ZX2IF	ZX1IF	ZXTO2IF

XXIF	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution Be sure to clear bit 2 of IF21, bits 3, 5 of IF22, bits 5 to 7 of IF23 to 0.

(4) Extended SFR (3rd SFR) Interrupt mask flag registers (MK20, MK21, MK22, MK23)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK20, MK21, MK22, and MK23 are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears these registers to FFH.

Figure 18-5. Format of Extended SFR (3rd SFR) Interrupt Mask Flag Registers (MK20, MK21, MK22, MK23)

Address: 84H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK20	REANOLDMK	APPNOLDMK	ACTSIGNMK	REASIGNMK	FAULTSIGNMK	ADMK2	RTC2MK	RTC0MK

Address: 85H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK21	REAEHFMK	APPEHFMK	ACTEOFMK	REAEOFMK	APPEOFMK	1	CFMK	ACTNOLDMK

Address: 86H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK22	ZXTO1MK	CYCENDMK	1	PKV1MK	1	PKI1MK	WFSMMK	ACTEHFMK

Address: 87H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK23	1	1	1	SAG2MK	SAG1MK	ZX2MK	ZX1MK	ZXTO2MK

XXMK	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bit 2 of MK21, bits 3, 5 of MK22, bits 5 to 7 of MK23 to 1.

(5) Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority level.

A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L, or 2H).

PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H can be set by a 1-bit or 8-bit memory manipulation instruction. If PR00L and PR00H, PR01L and PR01H, PR02L and PR02H, PR10L and PR10H, PR11L and PR11H, and PR12L and PR12H 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 18-6. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H (1/2)

Address: FFFE8H After reset: FFH R/W

Symbol	7	6	<5>	4	3	<2>	<1>	<0>
PR00L	1	1	PPR03	1	1	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	7	6	<5>	4	3	<2>	<1>	<0>
PR10L	1	1	PPR13	1	1	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00	SRPR00	STPR00	DMAPR01	DMAPR00	SREPR03	SRPR03	STPR03

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10	SRPR10	STPR10	DMAPR11	DMAPR10	SREPR13	SRPR13	STPR13

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICAPR0	1	1	CSIPR010

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICAPR1	1	1	CSIPR110

Caution Be sure to set bits 3, 4, 6, 7 of PR00L and PR10L, bits 1 to 3 of PR01L and PR11L to 1.

Figure 18-6. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (2/2)

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR01H	TMPR004	SREPR02	SRPR02	CSIPR020 IICPR020 STPR02	1	RTCIPR0	RTCPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR11H	TMPR104	SREPR12	SRPR12	CSIPR120 IICPR120 STPR12	1	RTCIPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	7	<6>	<5>	4	<3>	<2>	<1>	<0>
PR02L	1	PPR09	PPR08	1	PPR06	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	7	<6>	<5>	4	<3>	<2>	<1>	<0>
PR12L	1	PPR19	PPR18	1	PPR16	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

<R>

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	0
PR02H	1	1	MDPR0	TMPR013	TMPR012	TMPR011	TMPR010	1

Address: FFFDDH After reset: FFH R/W

<R>

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	0
PR12H	1	1	MDPR1	TMPR113	TMPR112	TMPR111	TMPR110	1

XXPR1X	XXPR0X	Priority level selection
0	0	Specify level 0 (high priority level)
0	1	Specify level 1
1	0	Specify level 2
1	1	Specify level 3 (low priority level)

Caution Be sure to set bit 3 of PR01H and PR11H, bits 4, 7 of PR02L and PR12L, bits 0, 6, 7 of PR02H and PR12H to 1.

(6) External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)

These registers specify the valid edge for INTP0, INTP3, INTP6, INTP9.

EGP0, EGP1, EGN0, and EGN1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 18-7. Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	0	EGP6	0	0	EGP3	0	0	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	0	EGN6	0	0	EGN3	0	0	EGN0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	0	0	EGP9	0

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	0	0	EGN9	0

EGPn	EGNn	INTPn pin valid edge selection (n = 0, 3, 6, 9)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 18-5 shows the ports corresponding to EGPn and EGNn.

Table 18-5. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120	INTP0
EGP3	EGN3	P33	INTP3
EGP6	EGN6	P11	INTP6
EGP9	EGN9	P81	INTP9

Caution Select the port mode by clearing EGPn and EGNn to 0 because an edge may be detected when the external interrupt function is switched to the port function.

Remark n = 0, 3, 6, 9

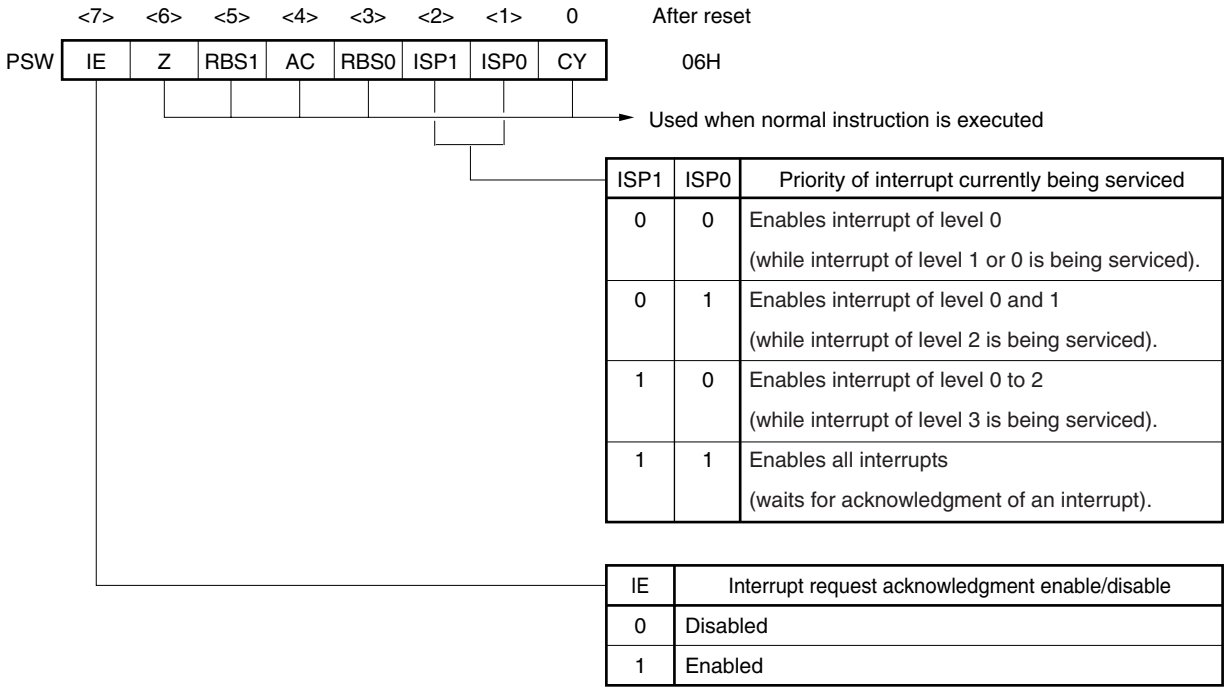
(7) Program status word (PSW)

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. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are 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 18-8. Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

18.4 Interrupt Servicing Operations

18.4.1 Maskable interrupt acknowledgment

A maskable interrupt 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 18-6 below.

For the interrupt request acknowledgment timing, see **Figures 18-10** and **18-11**.

Table 18-6. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
Servicing time	9 clocks	14 clocks

Note If an interrupt request is generated just before the RET instruction, the wait time becomes longer.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

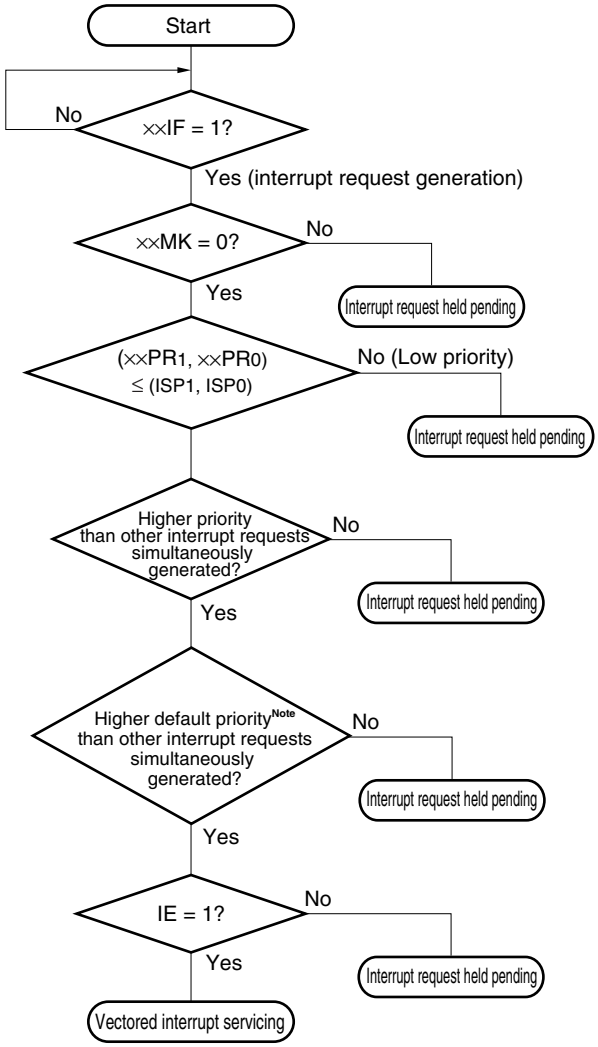
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 18-9 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.

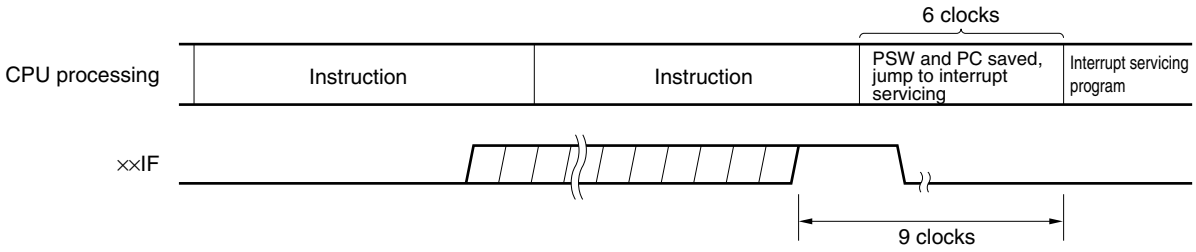
Figure 18-9. Interrupt Request Acknowledgment Processing Algorithm



××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR0: Priority specification flag 0
××PR1: Priority specification flag 1
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1: Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 18-8**)

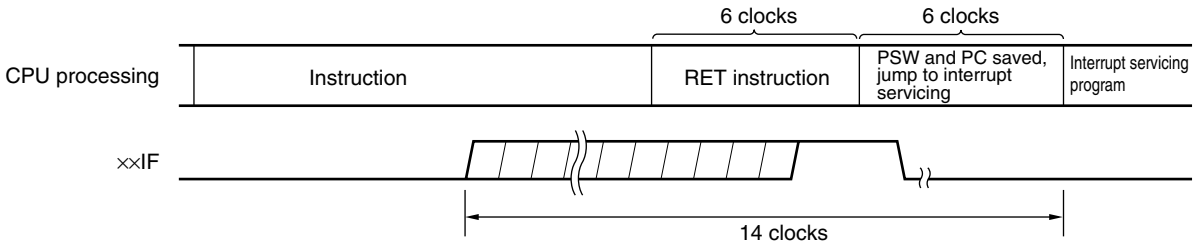
Note For the default priority, refer to **Table 18-1 Interrupt Source List**.

Figure 18-10. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

Figure 18-11. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

18.4.2 Software interrupt request acknowledgment

A software interrupt acknowledge 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 Do not use the RETI instruction for restoring from the software interrupt.

18.4.3 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 18-7 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 18-12 shows multiple interrupt servicing examples.

Table 18-7. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0, ISP0 = 0	○	×	×	×	×	×	×	×	○
	ISP1 = 0, ISP0 = 1	○	×	○	×	×	×	×	×	○
	ISP1 = 1, ISP0 = 0	○	×	○	×	○	×	×	×	○
	ISP1 = 1, ISP0 = 1	○	×	○	×	○	×	○	×	○
Software interrupt		○	×	○	×	○	×	○	×	○

Remarks 1. ○: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

3. ISP0, ISP1, and IE are flags contained in the PSW.

ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.

ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.

ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.

ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment.

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

4. PR is a flag contained in PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)

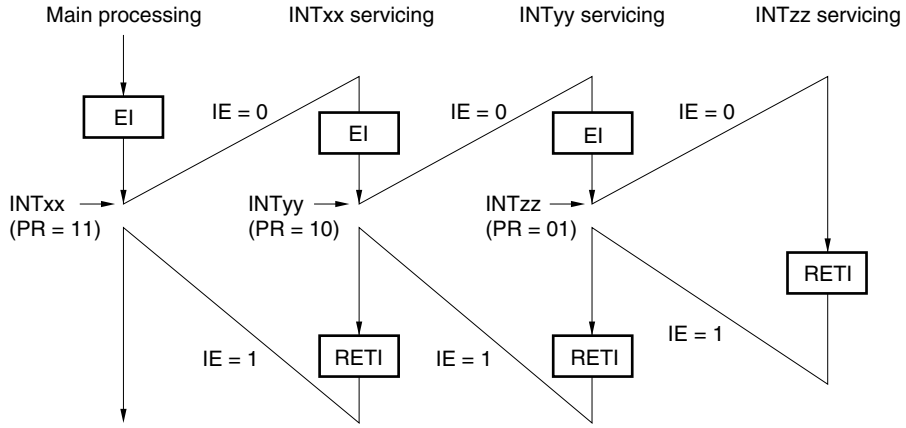
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 1 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

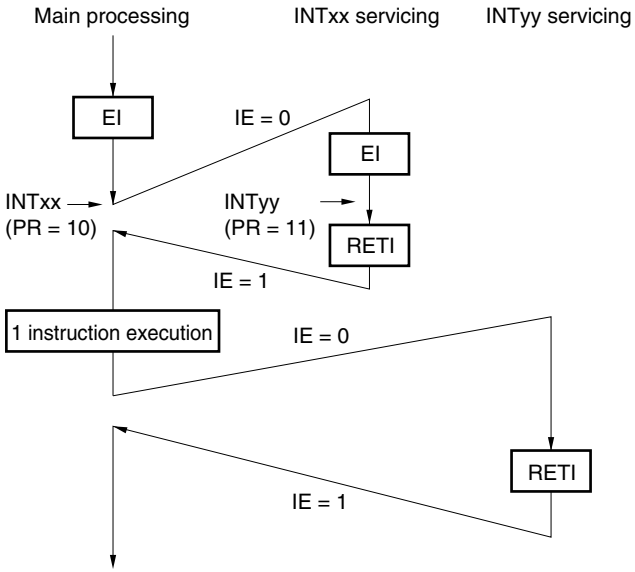
Figure 18-12. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

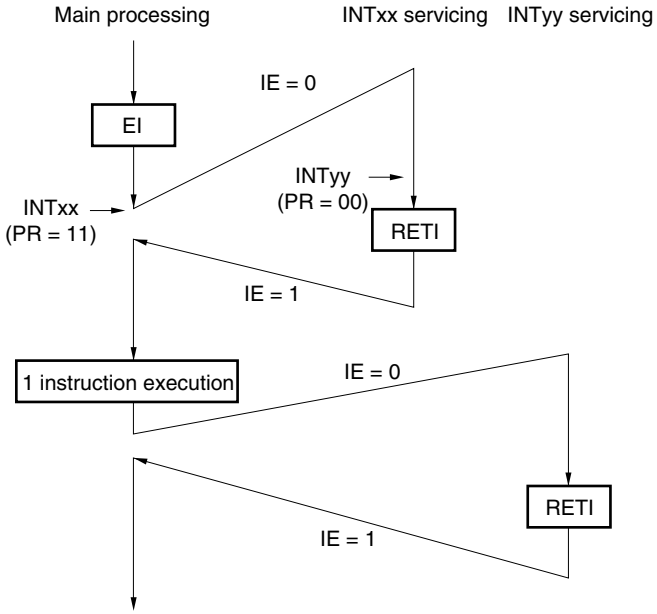


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 1 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 18-12. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 1 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

18.4.4 Interrupt request hold

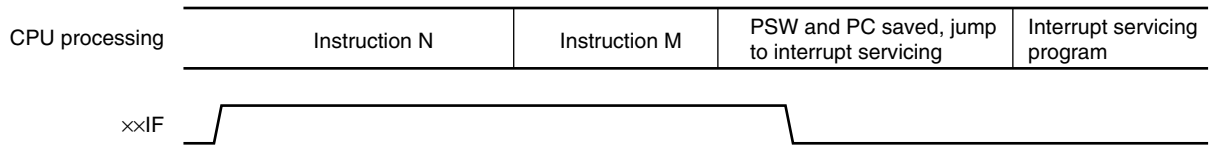
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr8
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- Manipulation instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers.

Caution The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged.

Figure 18-13 shows the timing at which interrupt requests are held pending.

Figure 18-13. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

CHAPTER 19 POWER CALCULATION CIRCUIT

19.1 Power Calculation Circuit Functions

This circuit measures the power and amount of power (energy) through linkage with the digital filter for the 24-bit ΔΣ-type A/D converter, and other functions (such as zero crossing, period and frequency measurement, SAG detection, and peak detection). The measurement functions include active power and energy calculation, reactive power and energy calculation, apparent power and energy calculation, and current and voltage RMS calculation.

The power calculation circuit can be used in both the single-phase two-wire and single-phase three-wire modes.

The power calculation circuit has the following functions:

- Integrator (for a di/dt current sensor (Rogowski coil))
- 90° phase shift (for reactive power calculation)
- Current and voltage RMS calculation
- Active power and energy calculation
- Reactive power and energy calculation
- Apparent power and energy calculation
- Ampere-hour accumulation
- Waveform sampling function

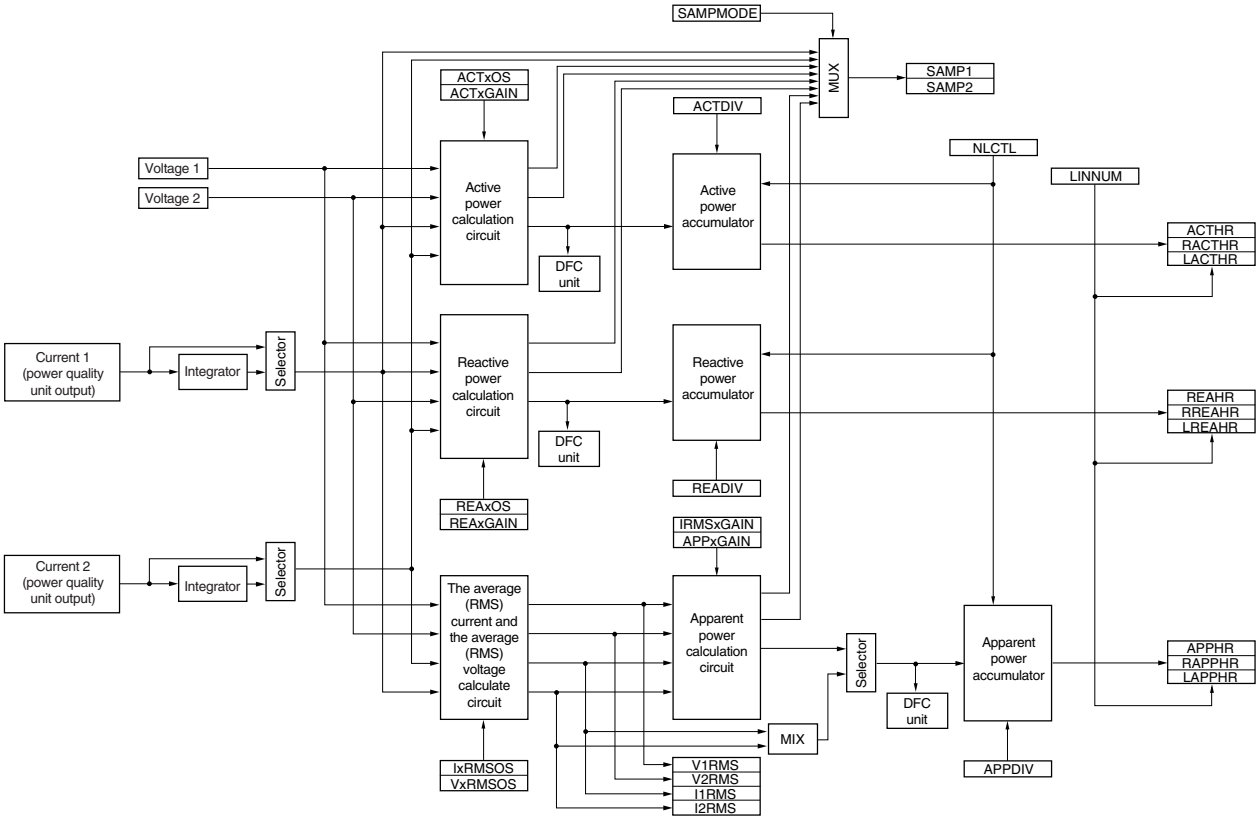
19.2 Configuration of Power Calculation Circuit

The power calculation circuit includes the following hardware.

Table 19-1. Configuration of Power Calculation Circuit

Item	Configuration
Controller	Current integrator circuit
	90° phase shifter circuit
	Current and voltage RMS calculation circuit
	Active power and energy calculation circuit
	Reactive power and energy calculation circuit
	Apparent power and energy calculation circuit
	Ampere-hour calculation circuit
	Waveform sampling circuit
Registers	RMS registers for voltage channels 1 and 2 (V1RMS, V2RMS)
	RMS registers for current channels 1 and 2 (I1RMS, I2RMS)
	Active power accumulation reading register (ACTHR)
	Active power accumulation reading and resetting register (RACTHR)
	Active power accumulation synchronous reading register (LACTHR)
	Reactive power accumulation reading register (REAHR)
	Reactive power accumulation reading and resetting register (RREAHR)
	Reactive power accumulation synchronous reading register (LREAHR)
	Apparent power accumulation reading register (APPHR)
	Apparent power accumulation reading and resetting register (RAPPHR)
	Apparent power accumulation synchronous reading register (LAPPHR)
	Sampling result registers 1 and 2 (SAMP1, SAMP2)
Control registers	Power calculation mode control register 1 (PWCTL1)
	Power calculation mode control register 2 (PWCTL2)
	No-load level control register (NLCTL)
	Active power scaling specification register (ACTDIV)
	Reactive power scaling specification register (READIV)
	Apparent power scaling specification register (APPDIV)
	Line cycle number specification register (LINNUM)
	Active power gain specification registers 1 and 2 (ACT1GAIN, ACT2GAIN)
	Reactive power gain specification registers 1 and 2 (REA1GAIN, REA2GAIN)
	Apparent power gain specification registers 1 and 2 (APP1GAIN, APP2GAIN)
	RMS gain specification registers for current channels 1 and 2 (IRMS1GAIN, IRMS2GAIN)
	Active power offset specification registers 1 and 2 (ACT1OS, ACT2OS)
	Reactive power offset specification registers 1 and 2 (REA1OS, REA2OS)
	RMS offset specification registers for current channels 1 and 2 (I1RMSOS, I2RMSOS)
	RMS offset specification registers for voltage channels 1 and 2 (V1RMSOS, V2RMSOS)
	Sampling mode selection register (SAMPMODE)

Figure 19-1. Block Diagram of Power Calculation Circuit



(1) Current integrator circuit

If a di/dt current sensor (such as a Rogowski coil) is connected, this circuit is used to integrate the current value.

(2) 90° phase shifter circuit

This circuit shifts the phase of the voltage channel 90° to calculate the reactive power.

(3) Current and voltage RMS calculation circuit

This circuit calculates the average (RMS) current and the average (RMS) voltage.

(4) Active power and energy calculation circuit

This circuit is used to calculate the active power and energy.

(5) Reactive power and energy calculation circuit

This circuit is used to calculate the reactive power and energy.

(6) Apparent power and energy calculation circuit

This circuit is used to calculate the apparent power and energy. Because the same circuit is used for this circuit and the ampere-hour calculation circuit, select and use one.

(7) Ampere-hour calculation circuit

This circuit is used to calculate the ampere-hour charge. Because the same circuit is used for this circuit and the apparent power and energy calculation circuit, select and use one.

(8) Waveform sampling circuit

This circuit is used to read the calculation results each time 24-bit ΔΣ-type A/D converter sampling is performed (including the current value, voltage value, active power, reactive power, and apparent power). Of these results, two can be selected using the SAMMODE register and read. For details, see **19.4.8 Waveform sampling function**.

19.3 Power Calculation Circuit

The power calculation circuit uses the following registers.

These registers are all allocated to the extended SFR (3rd SFR) space.

For details about how to access the extended SFR (3rd SFR) space, see **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE**.

- Power calculation mode control register 1 (PWCTL1)
- Power calculation mode control register 2 (PWCTL2)
- No-load level control register (NLCTL)
- Active power scaling specification register (ACTDIV)
- Reactive power scaling specification register (READIV)
- Apparent power scaling specification register (APPDIV)
- RMS registers for voltage channels 1 and 2 (V1RMS, V2RMS)
- RMS registers for current channels 1 and 2 (I1RMS, I2RMS)
- Active power accumulation reading register (ACTHR)
- Active power accumulation reading and resetting register (RACTHR)
- Active power accumulation synchronous reading register (LACTHR)
- Reactive power accumulation reading register (REAHR)
- Reactive power accumulation reading and resetting register (RREAHR)
- Reactive power accumulation synchronous reading register (LREAHR)
- Apparent power accumulation reading register (APPHR)
- Apparent power accumulation reading and resetting register (RAPPHR)
- Apparent power accumulation synchronous reading register (LAPPHR)
- Line cycle number specification register (LINNUM)
- Active power gain specification registers 1 and 2 (ACT1GAIN, ACT2GAIN)
- Reactive power gain specification registers 1 and 2 (REA1GAIN, REA2GAIN)
- Apparent power gain specification registers 1 and 2 (APP1GAIN, APP2GAIN)
- RMS gain specification registers for current channels 1 and 2 (IRMS1GAIN, IRMS2GAIN)
- Active power offset specification registers 1 and 2 (ACT1OS, ACT2OS)
- Reactive power offset specification registers 1 and 2 (REA1OS, REA2OS)
- RMS offset specification registers for current channels 1 and 2 (I1RMSOS, I2RMSOS)
- RMS offset specification registers for voltage channels 1 and 2 (V1RMSOS, V2RMSOS)
- Sampling mode selection register (SAMPMODE)
- Sampling result registers 1 and 2 (SAMP1, SAMP2)

(1) Power calculation mode control register 1 (PWCTL1)

The PWCTL1 register is used to control the power calculation, power quality measurement, and digital frequency conversion.

PWCTL1 is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 19-2. Format of Power Calculation Mode Control Register 1 (PWCTL1) (1/2)

Address: 180H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PWCTL1	PWREN	READIS	APPDIS	CYCDIS	SAMPEN	APPRMS CFCON	INTE2	INTE1

PWREN[Note 1]	Control power calculation, power quality measurement, and digital frequency conversion
0	Disables operation of the power calculation circuit.
1	Enables operation of the power calculation circuit.

READIS	Control reactive power measurement
0	Enables reactive power measurement.
1	Disables reactive power measurement.

APPDIS	Control apparent power measurement
0	Enables apparent power measurement.
1	Disables apparent power measurement.

CYCDIS	Control synchronous reading interrupt (CYCEND) detection
0	Enables synchronous reading interrupt (CYCEND) detection.
1	Disables synchronous reading interrupt (CYCEND) detection[Note 2].

Notes 1. By setting ADCE2 bit of ADM2 register and PWREN bit of PWCTL1 register, the operation is controlled as follows.

ADCE2	PWREN	24-bit ΔΣ-type A/D converter (analog block and digital block)	power calculation, power quality measurement, and digital frequency conversion
0	0	Cannot operate	Cannot operate
0	1	Cannot operate	Cannot operate
1	0	Operable	Cannot operate
1	1	Operable	Operable

2. If CYCDIS is 1, the CYCEND interrupt is not generated when the number of half-line cycles reaches the value specified for the LINNUM register. In addition, the accumulation register is not cleared when the LINNUM register is written to.

Figure 19-2. Format of Power Calculation Mode Control Register 1 (PWCTL1) (2/2)

Address: 180H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PWCTL1	PWREN	READIS	APPDIS	CYCDIS	SAMPEN	APPRMS CFCON	INTE2	INTE1

SAMPEN	Control waveform sample mode
0	Disables waveform sample mode.
1	Enables waveform sample mode.

APPRMS CFCON	Select apparent power or the average (RMS) current Irms for the apparent power accumulation register and CF pulse output
0	If this is specified, the apparent power accumulation register accumulates the apparent power. If the CFSEL1 bit of the CFCTL register is set to 1, the apparent power is output as CF pulses. CHMD = 0 (two-wire mode): Apparent power 1 is output as CF pulses. CHMD = 1 (three-wire mode): Apparent power 1 and apparent power 2 are output as CF pulses.
1	If this is specified, the apparent power accumulation register accumulates the average (RMS) current Irms. If the CFSEL1 bit of the CFCTL register is set to 1, the Irms is output as CF pulses. CHMD = 0 (two-wire mode): I1rms is output as CF pulses. CHMD = 1 (three-wire mode): I1rms and I2rms are output as CF pulses.

INTE2	Control integrator 2 (for current channel 2)
0	Disables operation of the integrator 2.
1	Enables operation of the integrator 2.

INTE1	Control integrator 1 (for current channel 1)
0	Disables operation of the integrator 1.
1	Enables operation of the integrator 1.

(2) Power calculation mode control register 2 (PWCTL2)

The PWCTL2 register is used to control the power calculation, power quality measurement, and digital frequency conversion.

PWCTL2 is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 19-3. Format of Power Calculation Mode Control Register 2 (PWCTL2) (1/2)

Address: 181H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PWCTL2	ACTDIS	0	REASIGN	ACTSIGN	ABSVARM	SAVARM	POAM	ABSAM

ACTDIS	Control active power measurement
0	Enables active power measurement.
1	Disables active power measurement.

REASIGN	Specify the trigger for generating a reactive power sign change interrupt (INTREASIGN)
0	Generates an interrupt when the reactive power sign changes from positive to negative.
1	Generates an interrupt when the reactive power sign changes from negative to positive.

ACTSIGN	Specify the trigger for generating a active power sign change interrupt (INTACTSIGN)
0	Generates an interrupt when the active power sign changes from positive to negative.
1	Generates an interrupt when the active power sign changes from negative to positive.

ABSVARM Note 1	Control accumulation of the absolute value of the reactive power and pulse output
0	Disables the reactive-power absolute-value accumulation mode.
1	Enables the reactive-power absolute-value accumulation mode.

SAVARM Note 1	Specify the sign for the reactive power accumulation mode
0	Specifies that the sign does not depend on the active power sign.
1	Specifies that the sign depends on the active power sign

Caution Be sure to clear bit 6 to 0.

Notes 1. The integration method for the reactive power signal can be selected by specifying values for the SAVARM and ABSVARM bits.

SAVARM	ABSVARM	Reactive Power Accumulation Mode
0	0	Signed value accumulation mode
0	1	Absolute value accumulation mode
1	0	Anti-tamper accumulation mode Note 2
1	1	Absolute value accumulation mode

2. If the active power is positive, the reactive power is added to the reactive energy accumulator. If the active power is negative, the reactive power is subtracted from the reactive energy accumulator. This setting affects both the reactive power registers and pulse output.

Figure 19-3. Format of Power Calculation Mode Control Register 2 (PWCTL2) (2/2)

Address: 181H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PWCTL2	ACTDIS	0	REASIGN	ACTSIGN	ABSVARM	SAVARM	POAM	ABSAM

POAM[Note]	Control accumulation of the positive value of the active power and pulse output
0	Disables the active-power positive value accumulation mode.
1	Enables the active-power positive value accumulation mode.

ABSAM[Note]	Control accumulation of the absolute value of the active power and pulse output
0	Disables the active-power absolute-value accumulation mode.
1	Enables the active-power absolute-value accumulation mode.

Caution Be sure to clear bit 6 to 0.

Note The integration method for the active power signal can be selected by specifying values for the POAM and ABSAM bits.

POAM	ABSAM	Reactive Power Accumulation Mode
0	0	Signed value accumulation mode
0	1	Absolute value accumulation mode
1	0	Positive value accumulation mode
1	1	Absolute value accumulation mode

(3) No-load level control register (NLCTL)

The NLCTL register is used to set the no-load detection.

NLCTL is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 19-4. Format of No-load Level Control Register (NLCTL)

Address: 182H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NLCTL	DISREACMP	ZXRMS	APPNO LOAD1	APPNO LOAD0	REANO LOAD1	REANO LOAD0	ACTNO LOAD1	ACTNO LOAD0

DISREACMP	Specify whether to enable reactive power gain compensation according to the line frequency
0	Enables gain compensation.
1	Disables gain compensation.

ZXRMS	Specify whether to synchronize with the line frequency when storing the rms value
0	Does not synchronize with the line frequency.
1	Synchronizes with the line frequency.

APPNO LOAD1	APPNO LOAD0	Specify the no-load threshold value for the apparent power and IRMS
0	0	Disables no-load detection for the apparent power and IRMS.
0	1	Enables no-load detection for the apparent power and IRMS. Threshold value = 0.03% of full scale
1	0	Enables no-load detection for the apparent power and IRMS. Threshold value = 0.015% of full scale
1	1	Enables no-load detection for the apparent power and IRMS. Threshold value = 0.0075% of full scale

REANO LOAD1	REANO LOAD0	Specify the no-load threshold value for the reactive power
0	0	Disables no-load detection for the reactive power.
0	1	Enables no-load detection for the reactive power. Threshold value = 0.015% of full scale
1	0	Enables no-load detection for the reactive power. Threshold value = 0.0075% of full scale
1	1	Enables no-load detection for the reactive power. Threshold value = 0.0037% of full scale

ACTNO LOAD1	ACTNO LOAD0	Specify the no-load threshold value for the active power
0	0	Disables no-load detection for the active power.
0	1	Enables no-load detection for the active power. Threshold value = 0.015% of full scale
1	0	Enables no-load detection for the active power. Threshold value = 0.0075% of full scale
1	1	Enables no-load detection for the active power. Threshold value = 0.0037% of full scale

(4) Active power scaling specification register (ACTDIV)

The ACTDIV register is used to set the active power scaling specification.

ACTDIV is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 19-5. Format of Active Power Scaling Specification Register (ACTDIV)

Address: 183H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ACTDIV	ACTDIV7	ACTDIV6	ACTDIV5	ACTDIV4	ACTDIV3	ACTDIV2	ACTDIV1	ACTDIV0

ACTDIV7 to 0	Specify the value of active power scaling
00H to FFH	Active power scaling Value

(5) Reactive power scaling specification register (READIV)

The READIV register is used to set the reactive power scaling specification.

READIV is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 19-6. Format of Reactive Power Scaling Specification Register (READIV)

Address: 184H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
READIV	READIV7	READIV6	READIV5	READIV4	READIV3	READIV2	READIV1	READIV0

READIV7 to 0	Specify the value of reactive power scaling
00H to FFH	Reactive power scaling value

(6) Apparent power scaling specification register (APPDIV)

The APPDIV register is used to set the apparent power scaling specification.

APPDIV is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 19-7. Format of Apparent Power Scaling Specification Register (APPDIV)

Address: 185H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
APPDIV	APPDIV7	APPDIV6	APPDIV5	APPDIV4	APPDIV3	APPDIV2	APPDIV1	APPDIV0

APPDIV7 to 0	Specify the value of apparent power scaling
00H to FFH	Apparent power scaling value

(7) RMS registers for voltage channels 1 and 2 (V1RMS, V2RMS)

These registers are used to store the RMS measured value of voltage channels 1 and 2.

V1RMS and V2RMS are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 000000H.

Figure 19-8. Format of RMS Registers for Voltage Channels 1 and 2 (V1RMS, V2RMS)

(a) V1RMS

Address: 186H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
V1RMSL	V1RMS7	V1RMS6	V1RMS5	V1RMS4	V1RMS3	V1RMS2	V1RMS1	V1RMS0

Address: 187H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
V1RMSM	V1RMS15	V1RMS14	V1RMS13	V1RMS12	V1RMS11	V1RMS10	V1RMS9	V1RMS8

Address: 188H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
V1RMSH	V1RMS23	V1RMS22	V1RMS21	V1RMS20	V1RMS19	V1RMS18	V1RMS17	V1RMS16

(b) V2RMS

Address: 189H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
V2RMSL	V2RMS7	V2RMS6	V2RMS5	V2RMS4	V2RMS3	V2RMS2	V2RMS1	V2RMS0

Address: 18AH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
V2RMSM	V2RMS15	V2RMS14	V2RMS13	V2RMS12	V2RMS11	V2RMS10	V2RMS9	V2RMS8

Address: 18BH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
V2RMSH	V2RMS23	V2RMS22	V2RMS21	V2RMS20	V2RMS19	V2RMS18	V2RMS17	V2RMS16

VnRMS23 to 0	Used to store the RMS measured value of voltage channels n (n = 1, 2)
000000H to FFFFFFH	RMS measured value of voltage channels n

(8) RMS registers for current channels 1 and 2 (I1RMS, I2RMS)

These registers are used to store the RMS measured value of current channels 1 and 2.

I1RMS and I2RMS are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 000000H.

Figure 19-9. Format of RMS Registers for Current Channels 1 and 2 (I1RMS, I2RMS)

(a) I1RMS

Address: 18CH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
I1RMSL	I1RMS7	I1RMS6	I1RMS5	I1RMS4	I1RMS3	I1RMS2	I1RMS1	I1RMS0

Address: 18DH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
I1RMSM	I1RMS15	I1RMS14	I1RMS13	I1RMS12	I1RMS11	I1RMS10	I1RMS9	I1RMS8

Address: 18EH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
I1RMSH	I1RMS23	I1RMS22	I1RMS21	I1RMS20	I1RMS19	I1RMS18	I1RMS17	I1RMS16

(b) I2RMS

Address: 18FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
I2RMSL	I2RMS7	I2RMS6	I2RMS5	I2RMS4	I2RMS3	I2RMS2	I2RMS1	I2RMS0

Address: 190H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
I2RMSM	I2RMS15	I2RMS14	I2RMS13	I2RMS12	I2RMS11	I2RMS10	I2RMS9	I2RMS8

Address: 191H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
I2RMSH	I2RMS23	I2RMS22	I2RMS21	I2RMS20	I2RMS19	I2RMS18	I2RMS17	I2RMS16

InRMS23 to 0	Used to store the RMS measured value of current channels n (n = 1, 2)
000000H to FFFFFFH	RMS measured value of current channels n

(9) Active power accumulation reading register (ACTHR)

This register is used to store the active power accumulation value.
This register value is not cleared after being read.
ACTHR is allocated to the extended SFR (3rd SFR) space.
Use the extended SFR (3rd SFR) interface to set up this register.
Reset signal generation clears this register to 000000H.

Figure 19-10. Format of Active Power Accumulation Reading Register (ACTHR)

Address: 192H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ACTHRL	ACTHR7	ACTHR6	ACTHR5	ACTHR4	ACTHR3	ACTHR2	ACTHR1	ACTHR0

Address: 193H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ACTHRM	ACTHR15	ACTHR14	ACTHR13	ACTHR12	ACTHR11	ACTHR10	ACTHR9	ACTHR8

Address: 194H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ACTHRH	ACTHR23	ACTHR22	ACTHR21	ACTHR20	ACTHR19	ACTHR18	ACTHR17	ACTHR16

ACTHR23 to 0	Used to store the accumulator value of the active power
000000H to FFFFFFH	Accumulator value of the active power (this register value is not cleared after being read)

(10) Active power accumulation reading and resetting register (RACTHR)

The active power accumulation value is stored in this register.
This register value is cleared after being read.
RACTHR is allocated to the extended SFR (3rd SFR) space.
Use the extended SFR (3rd SFR) interface to set up this register.
Reset signal generation clears this register to 000000H.

Figure 19-11. Format of Active Power Accumulation Reading and Resetting Register (RACTHR)

Address: 196H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RACTHRL	RACTHR7	RACTHR6	RACTHR5	RACTHR4	RACTHR3	RACTHR2	RACTHR1	RACTHR0

Address: 197H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RACTHRM	RACTHR15	RACTHR14	RACTHR13	RACTHR12	RACTHR11	RACTHR10	RACTHR9	RACTHR8

Address: 198H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RACTHRH	RACTHR23	RACTHR22	RACTHR21	RACTHR20	RACTHR19	RACTHR18	RACTHR17	RACTHR16

RACTHR23 to 0	Used to store the accumulator value of the active power
000000H to FFFFFFH	Accumulator value of the active power (this register value is cleared after being read)

(11) Active power accumulation synchronous reading register (LACTHR)

This register synchronizes the accumulator value of the active power with the line frequency and is used to store the value.

LACTHR is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 000000H.

Figure 19-12. Format of Active Power Accumulation Synchronous Reading Register (LACTHR)

Address: 199H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
LACTHRL	LACTHR7	LACTHR6	LACTHR5	LACTHR4	LACTHR3	LACTHR2	LACTHR1	LACTHR0

Address: 19AH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
LACTHRM	LACTHR15	LACTHR14	LACTHR13	LACTHR12	LACTHR11	LACTHR10	LACTHR9	LACTHR8

Address: 19BH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
LACTHRH	LACTHR23	LACTHR22	LACTHR21	LACTHR20	LACTHR19	LACTHR18	LACTHR17	LACTHR16

LACTHR23 to 0	Used to store the accumulator value of the active power
000000H to FFFFFFH	Accumulator value of the active power (stored in the register after synchronization with the line frequency)

(12) Reactive power accumulation reading register (REAHR)

The reactive power accumulation value is stored in this register.

This register value is not cleared after being read.

REAHR is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 000000H.

Figure 19-13. Format of Reactive Power Accumulation Reading Register (REAHR)

Address: 19CH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
REAHRL	REAHR7	REAHR6	REAHR5	REAHR4	REAHR3	REAHR2	REAHR1	REAHR0

Address: 19DH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
REAHRM	REAHR15	REAHR14	REAHR13	REAHR12	REAHR11	REAHR10	REAHR9	REAHR8

Address: 19EH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
REAHRH	REAHR23	REAHR22	REAHR21	REAHR20	REAHR19	REAHR18	REAHR17	REAHR16

REAHR23 to 0	Used to store the accumulator value of the reactive power
000000H to FFFFFFH	Accumulator value of the reactive power (this register value is not cleared after being read)

(13) Reactive power accumulation reading and resetting register (RREAHR)

The reactive power accumulation value is stored in this register.
This register value is cleared after being read.
RREAHR is allocated to the extended SFR (3rd SFR) space.
Use the extended SFR (3rd SFR) interface to set up this register.
Reset signal generation clears this register to 000000H.

Figure 19-14. Format of Reactive Power Accumulation Reading and Resetting Register (RREAHR)

Address: 1A0H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RREAHRL	RREAHR7	RREAHR6	RREAHR5	RREAHR4	RREAHR3	RREAHR2	RREAHR1	RREAHR0

Address: 1A1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RREAHRM	RREAHR15	RREAHR14	RREAHR13	RREAHR12	RREAHR11	RREAHR10	RREAHR9	RREAHR8

Address: 1A2H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RREAHRH	RREAHR23	RREAHR22	RREAHR21	RREAHR20	RREAHR19	RREAHR18	RREAHR17	RREAHR16

RREAHR23 to 0	Used to store the accumulator value of the reactive power
000000H to FFFFFFH	Accumulator value of the reactive power (this register value is cleared after being read)

(14) Reactive power accumulation synchronous reading register (LREAHR)

This register synchronizes the accumulator value of the reactive power with the line frequency and is used to store the value.
LREAHR is allocated to the extended SFR (3rd SFR) space.
Use the extended SFR (3rd SFR) interface to set up this register.
Reset signal generation clears this register to 000000H.

Figure 19-15. Format of Reactive Power Accumulation Synchronous Reading Register (LREAHR)

Address: 1A3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
LREAHRL	LREAHR7	LREAHR6	LREAHR5	LREAHR4	LREAHR3	LREAHR2	LREAHR1	LREAHR0

Address: 1A4H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
LREAHRM	LREAHR15	LREAHR14	LREAHR13	LREAHR12	LREAHR11	LREAHR10	LREAHR9	LREAHR8

Address: 1A5H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
LREAHRH	LREAHR23	LREAHR22	LREAHR21	LREAHR20	LREAHR19	LREAHR18	LREAHR17	LREAHR16

LREAHR23 to 0	Used to store the accumulator value of the reactive power
000000H to FFFFFFH	Accumulator value of the reactive power (stored in the register after synchronization with the line frequency)

(15) Apparent power accumulation reading register (APPHR)

The apparent power accumulation value is stored in this register.
This register value is not cleared after being read.
APPHR is allocated to the extended SFR (3rd SFR) space.
Use the extended SFR (3rd SFR) interface to set up this register.
Reset signal generation clears this register to 000000H.

Figure 19-16. Format of Apparent Power Accumulation Reading Register (APPHR)

Address: 1A6H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
APPHRL	APPHR7	APPHR6	APPHR5	APPHR4	APPHR3	APPHR2	APPHR1	APPHR0

Address: 1A7H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
APPHRM	APPHR15	APPHR14	APPHR13	APPHR12	APPHR11	APPHR10	APPHR9	APPHR8

Address: 1A8H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
APPHRH	APPHR23	APPHR22	APPHR21	APPHR20	APPHR19	APPHR18	APPHR17	APPHR16

APPHR23 to 0	Used to store the accumulator value of the apparent power
000000H to FFFFFFH	Accumulator value of the apparent power (this register value is not cleared after being read)

(16) Apparent power accumulation reading and resetting register (RAPPHR)

The reactive power accumulation value is stored in this register.
This register value is cleared after being read.
RAPPHR is allocated to the extended SFR (3rd SFR) space.
Use the extended SFR (3rd SFR) interface to set up this register.
Reset signal generation clears this register to 000000H.

Figure 19-17. Format of Apparent Power Accumulation Reading and Resetting Register (RAPPHR)

Address: 1AAH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RAPPHRL	RAPPHR7	RAPPHR6	RAPPHR5	RAPPHR4	RAPPHR3	RAPPHR2	RAPPHR1	RAPPHR0

Address: 1ABH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RAPPHRM	RAPPHR15	RAPPHR14	RAPPHR13	RAPPHR12	RAPPHR11	RAPPHR10	RAPPHR9	RAPPHR8

Address: 1ACH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RAPPHRH	RAPPHR23	RAPPHR22	RAPPHR21	RAPPHR20	RAPPHR19	RAPPHR18	RAPPHR17	RAPPHR16

RAPPHR23 to 0	Used to store the accumulator value of the apparent power
000000H to FFFFFFH	Accumulator value of the apparent power (this register value is cleared after being read)

(17) Apparent power accumulation synchronous reading register (LAPPHR)

This register synchronizes the accumulator value of the apparent power with the line frequency and is used to store the value.

LAPPH is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 000000H.

Figure 19-18. Format of Apparent Power Accumulation Synchronous Reading Register (LAPPHR)

Address: 1ADH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
LAPPHRL	LAPPHR7	LAPPHR6	LAPPHR5	LAPPHR4	LAPPHR3	LAPPHR2	LAPPHR1	LAPPHR0

Address: 1AEH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
LAPPHRM	LAPPHR15	LAPPHR14	LAPPHR13	LAPPHR12	LAPPHR11	LAPPHR10	LAPPHR9	LAPPHR8

Address: 1AFH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
LAPPHRH	LAPPHR23	LAPPHR22	LAPPHR21	LAPPHR20	LAPPHR19	LAPPHR18	LAPPHR17	LAPPHR16

LAPPHR23 to 0	Used to store the accumulator value of the apparent power
000000H to FFFFFFH	Accumulator value of the apparent power (stored in the register after synchronization with the line frequency)

(18) Line cycle number specification register (LINNUM)

The LINNUM register is used to set the line cycle number of accumulator in synchronous reading mode.

This register can be set by a half line unit.

LINNUM is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to FFFFH.

Figure 19-19. Format of Line Cycle Number Specification Register (LINNUM)

Address: 1B0H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
LINNUML	LINNUM7	LINNUM6	LINNUM5	LINNUM4	LINNUM3	LINNUM2	LINNUM1	LINNUM0

Address: 1B1H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
LINNUMH	LINNUM15	LINNUM14	LINNUM13	LINNUM12	LINNUM11	LINNUM10	LINNUM9	LINNUM8

LINNUM15 to 0	Specify the half line cycle number of accumulator
0000H to FFFFH	Half line cycle number

(19) Active power gain specification registers 1 and 2 (ACT1GAIN, ACT2GAIN)

The ACT1GAIN and ACT2GAIN registers are used to set the gain value of the active power.

ACT1GAIN and ACT2GAIN are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 0000H.

Figure 19-20. Format of Active Power Gain Specification Registers 1 and 2 (ACT1GAIN, ACT2GAIN)

(a) ACT1GAIN

Address: 1B2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ACT1GAINL	ACT1GAIN7	ACT1GAIN6	ACT1GAIN5	ACT1GAIN4	ACT1GAIN3	ACT1GAIN2	ACT1GAIN1	ACT1GAIN0

Address: 1B3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ACT1GAINH	0	0	0	0	ACT1GAIN11	ACT1GAIN10	ACT1GAIN9	ACT1GAIN8

(b) ACT2GAIN

Address: 1B4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ACT2GAINL	ACT2GAIN7	ACT2GAIN6	ACT2GAIN5	ACT2GAIN4	ACT2GAIN3	ACT2GAIN2	ACT2GAIN1	ACT2GAIN0

Address: 1B5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ACT2GAINH	0	0	0	0	ACT2GAIN11	ACT2GAIN10	ACT2GAIN9	ACT2GAIN8

ACTnGAIN11 to 0	Specify the gain value of the active power (n = 1, 2)
0000H to 0FFFH	Gain value of active power

Caution Be sure to clear bits 4 to 7 of the ACT1GAINH and ACT2GAINH to 0.

(20) Reactive power gain specification registers 1 and 2 (REA1GAIN, REA2GAIN)

The REA1GAIN and REA2GAIN registers are used to set the gain value of the reactive power.

REA1GAIN and REA2GAIN are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 0000H.

Figure 19-21. Format of Reactive Power Gain Specification Registers 1 and 2 (REA1GAIN, REA2GAIN)

(a) REA1GAIN

Address: 1B6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
REA1GAINL	REA1GAIN7	REA1GAIN6	REA1GAIN5	REA1GAIN4	REA1GAIN3	REA1GAIN2	REA1GAIN1	REA1GAIN0

Address: 1B7H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
REA1GAINH	0	0	0	0	REA1GAIN11	REA1GAIN10	REA1GAIN9	REA1GAIN8

(b) REA2GAIN

Address: 1B8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
REA2GAINL	REA2GAIN7	REA2GAIN6	REA2GAIN5	REA2GAIN4	REA2GAIN3	REA2GAIN2	REA2GAIN1	REA2GAIN0

Address: 1B9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
REA2GAINH	0	0	0	0	REA2GAIN11	REA2GAIN10	REA2GAIN9	REA2GAIN8

REAnGAIN11 to 0	Specify the gain value of the reactive power (n = 1, 2)
0000H to 0FFFH	Gain value of reactive power

Caution Be sure to clear bits 4 to 7 of the REA1GAINH and REA2GAINH to 0.

(21) Apparent power gain specification registers 1 and 2 (APP1GAIN, APP2GAIN)

The APP1GAIN and APP2GAIN registers are used to set the gain value of apparent power.

APP1GAIN and APP2GAIN are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 0000H.

Figure 19-22. Format of Apparent Power Gain Specification Registers 1 and 2 (APP1GAIN, APP2GAIN)

(a) APP1GAIN

Address: 1BAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
APP1GAINL	APP1GAIN7	APP1GAIN6	APP1GAIN5	APP1GAIN4	APP1GAIN3	APP1GAIN2	APP1GAIN1	APP1GAIN0

Address: 1BBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
APP1GAINH	0	0	0	0	APP1GAIN11	APP1GAIN10	APP1GAIN9	APP1GAIN8

(b) APP2GAIN

Address: 1BCH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
APP2GAINL	APP2GAIN7	APP2GAIN6	APP2GAIN5	APP2GAIN4	APP2GAIN3	APP2GAIN2	APP2GAIN1	APP2GAIN0

Address: 1BDH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
APP2GAINH	0	0	0	0	APP2GAIN11	APP2GAIN10	APP2GAIN9	APP2GAIN8

APPnGAIN11 to 0	Specify the gain value of the apparent power (n = 1, 2)
0000H to 0FFFH	Gain value of apparent power

Caution Be sure to clear bits 4 to 7 of the APP1GAINH and APP2GAINH to 0.

(22) RMS gain specification registers for current channels 1 and 2 (IRMS1GAIN, IRMS2GAIN)

The IRMS1GAIN and IRMS2GAIN registers are used to set the gain value of the current power.

IRMS1GAIN and IRMS2GAIN are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 0000H.

Figure 19-23. Format of RMS gain Specification Registers for Current Channels 1 and 2 (IRMS1GAIN, IRMS2GAIN)

(a) IRMS1GAIN

Address: 1BEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IRMS1 GAINL	IRMS1 GAIN7	IRMS1 GAIN6	IRMS1 GAIN5	IRMS1 GAIN4	IRMS1 GAIN3	IRMS1 GAIN2	IRMS1 GAIN1	IRMS1 GAIN0

Address: 1BFH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IRMS1 GAINH	0	0	0	0	IRMS1 GAIN11	IRMS1 GAIN10	IRMS1 GAIN9	IRMS1 GAINN8

(b) IRMS2GAIN

Address: 1C0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IRMS2 GAINL	IRMS2 GAIN7	IRMS2 GAIN6	IRMS2 GAIN5	IRMS2 GAIN4	IRMS2 GAIN3	IRMS2 GAIN2	IRMS2 GAIN1	IRMS2 GAIN0

Address: 1C1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IRMS2 GAINH	0	0	0	0	IRMS2 GAIN11	IRMS2 GAIN10	IRMS2 GAIN9	IRMS2 GAIN8

IRMSnGAIN11 to 0	Specify the RMS gain value of the current power (n = 1, 2)
0000H to 0FFFH	RMS gain value of apparent power

Caution Be sure to clear bits 4 to 7 of the IRMS1GAINH and IRMS2GAINH to 0.

(23) Active power offset specification registers 1 and 2 (ACT1OS, ACT2OS)

The ACT1OS and ACT2OS registers are used to set the offset value of the active power.

ACT1OS and ACT2OS are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 0000H.

Figure 19-24. Format of Active Power Offset Specification Registers 1 and 2 (ACT1OS, ACT2OS)

(a) ACT1OS

Address: 1C2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ACT1OSL	ACT1OS7	ACT1OS6	ACT1OS5	ACT1OS4	ACT1OS3	ACT1OS2	ACT1OS1	ACT1OS0

Address: 1C3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ACT1OSH	ACT1OS15	ACT1OS14	ACT1OS13	ACT1OS12	ACT1OS11	ACT1OS10	ACT1OS9	ACT1OS8

(b) ACT2OS

Address: 1C4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ACT2OSL	ACT2OS7	ACT2OS6	ACT2OS5	ACT2OS4	ACT2OS3	ACT2OS2	ACT2OS1	ACT2OS0

Address: 1C5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ACT2OSH	ACT2OS15	ACT2OS14	ACT2OS13	ACT2OS12	ACT2OS11	ACT2OS10	ACT2OS9	ACT2OS8

ACTnOS15 to 0	Specify the offset value of the active power (n = 1, 2)
0000H to FFFFH	Offset value of active power

(24) Reactive power offset specification registers 1 and 2 (REA1OS, REA2OS)

The REA1OS and REA1OS registers are used to set the offset value of the reactive power.

REA1OS and REA1OS are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 0000H.

Figure 19-25. Format of Reactive power offset specification registers 1 and 2 (REA1OS, REA2OS)

(a) REA1OS

Address: 1C6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
REA1OSL	REA1OS7	REA1OS6	REA1OS5	REA1OS4	REA1OS3	REA1OS2	REA1OS1	REA1OS0

Address: 1C7H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
REA1OSH	REA1OS15	REA1OS14	REA1OS13	REA1OS12	REA1OS11	REA1OS10	REA1OS9	REA1OS8

(b) REA2OS

Address: 1C8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
REA2OSL	REA2OS7	REA2OS6	REA2OS5	REA2OS4	REA2OS3	REA2OS2	REA2OS1	REA2OS0

Address: 1C9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
REA2OSH	REA2OS15	REA2OS14	REA2OS13	REA2OS12	REA2OS11	REA2OS10	REA2OS9	REA2OS8

REAnOS15 to 0	Specify the offset value of the reactive power (n = 1, 2)
0000H to FFFFH	Offset value of reactive power

(25) RMS offset specification registers for current channels 1 and 2 (I1RMSOS, I2RMSOS)

The I1RMSOS and I2RMSOS registers are used to set the offset value of current channels 1 and 2.

I1RMSOS and I2RMSOS are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 0000H.

Figure 19-26. Format of RMS Offset Specification Registers for Current Channels 1 and 2 (I1RMSOS, I2RMSOS)

(a) I1RMSOS

Address: 1CAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
I1RMSOSL	I1RMSOS7	I1RMSOS6	I1RMSOS5	I1RMSOS4	I1RMSOS3	I1RMSOS2	I1RMSOS1	I1RMSOS0

Address: 1CBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
I1RMSOSH	0	0	0	0	I1RMSOS11	I1RMSOS10	I1RMSOS9	I1RMSOS8

(b) I2RMSOS

Address: 1CEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
I2RMSOSL	I2RMSOS7	I2RMSOS6	I2RMSOS5	I2RMSOS4	I2RMSOS3	I2RMSOS2	I2RMSOS1	I2RMSOS0

Address: 1CFH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
I2RMSOSH	0	0	0	0	I2RMSOS11	I2RMSOS10	I2RMSOS9	I2RMSOS8

InRMSOS11 to 0	Specify the RMS offset value of the current channel n (n = 1, 2)
0000H to 0FFFH	RMS offset value of current channel n

Caution Be sure to clear bits 4 to 7 of the I1RMSOSH and I2RMSOSH to 0.

(26) RMS offset specification registers for voltage channels 1 and 2 (V1RMSOS, V2RMSOS)

The V1RMSOS and V2RMSOS registers are used to set the offset value of voltage channels 1 and 2.

V1RMSOS and V2RMSOS are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 0000H.

Figure 19-27. Format of RMS Offset Specification Registers for Voltage Channels 1 and 2 (V1RMSOS, V2RMSOS)

(a) V1RMSOS

Address: 1CCH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
V1RMSOSL	V1RMSOS7	V1RMSOS6	V1RMSOS5	V1RMSOS4	V1RMSOS3	V1RMSOS2	V1RMSOS1	V1RMSOS0

Address: 1CDH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
V1RMSOSH	0	0	0	0	V1RMSOS11	V1RMSOS10	V1RMSOS9	V1RMSOS8

(b) V2RMSOS

Address: 1D0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
V2RMSOSL	V2RMSOS7	V2RMSOS6	V2RMSOS5	V2RMSOS4	V2RMSOS3	V2RMSOS2	V2RMSOS1	V2RMSOS0

Address: 1D1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
V2RMSOSH	0	0	0	0	V2RMSOS11	V2RMSOS10	V2RMSOS9	V2RMSOS8

VnRMSOS11 to 0	Specify the RMS offset value of voltage channel n (n = 1, 2)
0000H to 0FFFH	RMS offset value of voltage channel n

Caution Be sure to clear bits 4 to 7 of the V1RMSOSH and V2RMSOSH to 0.

(27) Sampling mode selection register (SAMPMODE)

The SAMPMODE register is used to select the sampling waveform.

SAMPMODE is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 19-28. Format of Sampling Mode Selection Register (SAMPMODE)

Address: 1D2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SAMPMODE	0	SAMP2SEL2	SAMP2SEL1	SAMP2SEL0	0	SAMP1SEL2	SAMP1SEL1	SAMP1SEL0

SAMPnSEL2	SAMPnSEL1	SAMPnSEL0	Select the waveform n for sample mode (n = 1, 2)
0	0	0	I1 output from integrator
0	0	1	I2 output from integrator
0	1	0	Active power 1 (output of ACT1GAIN after multiplication)
0	1	1	Active power 2 (output of ACT2GAIN after multiplication)
1	0	0	Reactive power 1 (output of REA1GAIN after multiplication)
1	0	1	Reactive power 2 (output of REA2GAIN after multiplication)
1	1	0	Apparent power 1 (output of APP1GAIN after multiplication)
1	1	1	Apparent power 2 (output of APP2GAIN after multiplication)

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

(28) Sampling result registers 1 and 2 (SAMP1, SAMP2)

The calculation result by sampling is stored in these registers.

SAMP1 and SAMP2 are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 000000H.

Figure 19-29. Format of Sampling Result Registers 1 and 2 (SAMP1, SAMP2)

(a) SAMP1

Address: 1D3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
SAMP1L	SAMP17	SAMP16	SAMP15	SAMP14	SAMP13	SAMP12	SAMP11	SAMP10

Address: 1D4H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
SAMP1M	SAMP115	SAMP114	SAMP113	SAMP112	SAMP111	SAMP110	SAMP19	SAMP18

Address: 1D5H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
SAMP1H	SAMP123	SAMP122	SAMP121	SAMP120	SAMP119	SAMP118	SAMP117	SAMP116

(b) SAMP2

Address: 1D6H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
SAMP2L	SAMP27	SAMP26	SAMP25	SAMP24	SAMP23	SAMP22	SAMP21	SAMP20

Address: 1D7H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
SAMP2M	SAMP215	SAMP214	SAMP213	SAMP212	SAMP211	SAMP210	SAMP29	SAMP28

Address: 1D8H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
SAMP2H	SAMP223	SAMP222	SAMP221	SAMP220	SAMP219	SAMP218	SAMP217	SAMP216

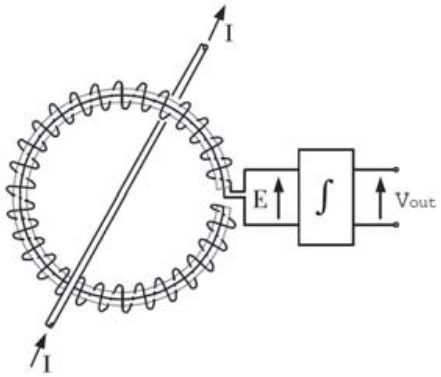
SAMPn23 to n0	Used to store the calculation result by sampling (n = 1, 2)
000000H to FFFFFFH	Calculation result by sampling

19.4 Power and Energy Calculation

19.4.1 Current integrator

If using a di/dt sensor, the current integrator is used to integrate the current. di/dt sensors, such as Rogowski coils, are systems that measure the primary circuit current and secondary circuit current (induced electromotive force) by using mutual inductance. When the primary circuit current flows, a magnetic field is generated around it, and this field is converted to current by the coil. This secondary circuit current shows how much the primary circuit current changes (di/dt).

Figure 19-30. Rogowski Coils



If using a Rogowski coil as a current sensor, the input current signal is a di/dt signal, and, because the current signal must be restored before the power can be calculated, an integrator is included.

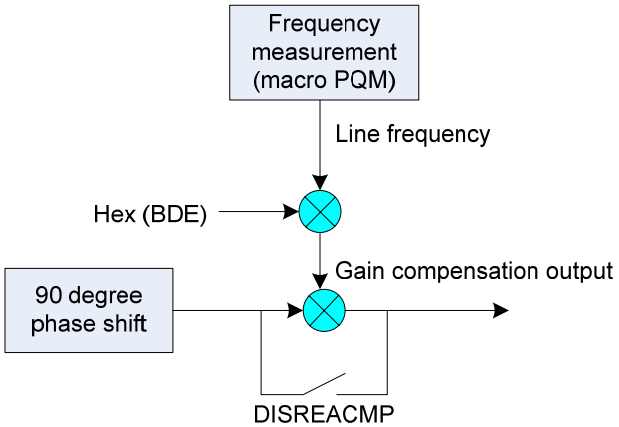
There are two integrators, integrator 1 for current channel 1 and integrator 2 for current channel 2. During a reset, the integrators for the current channels are turned off. To turn the integrators on, set the INTE1 and INTE2 bits of the PWCTL1 register to 1.

19.4.2 90° phase shifter

During reactive power or energy calculation, a 90° phase shift is needed between the current and voltage channel. The phase-shifting filter is a single-pole low-pass filter that performs a 90° phase shift over the relevant frequency (50 Hz or 60 Hz) and performs attenuation of 20 dB/dec.

A gain compensator is placed after the 90° phase shift. For a 90° phase shift, the frequency changes –20 dB/dec, so gain compensation is performed by multiplying the line frequency with the 90° phase shift output. However, gain compensation is only effective at the line frequency. The line frequency is measured by using the power quality measurement circuit. Gain compensation can be disabled using the DISREACMP bit of the NLCTL register.

Figure 19-31. 90° phase Shifter



19.4.3 Current and voltage RMS calculation

(1) RMS calculation method

The root mean square (rms) value of a signal V (t) is defined as:

$$Vrms = \sqrt{\frac{1}{T} * \int_0^T V^2(t)dt}$$

For sampled signals, RMS calculation involves squaring the signal, obtaining the average, and calculating the square root. This average is calculated by implementing a low-pass filter (LPF3). This low-pass filter has a –3 dB cutoff frequency of about 1.3 Hz when the sampling frequency is 4.34 kHz.

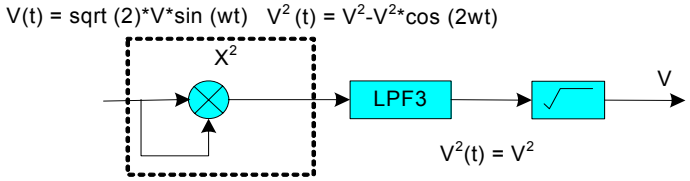
$$V(t) = \sqrt{2} \times V \sin(\omega t)$$

Here, V is the RMS voltage.

$$V^2(t) = V^2 - V^2 \cos(2\omega t)$$

When this signal goes through LPF3, the cos (2wt) term is attenuated and only the DC term $Vrms^2$ goes through.

Figure 19-32. Low-pass Filter (LPF3)



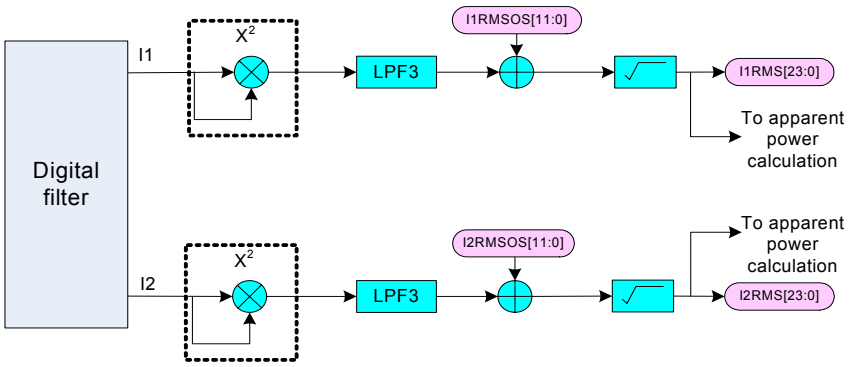
Four RMS values must be calculated: I1rms, I2rms, V1rms, and V2rms.
The RMS values can be read from the 24-bit registers I1RMS, I2RMS, V1RMS, and V2RMS, respectively.

(2) Current channel RMS calculation

This circuit calculates the RMS values for the two input current channels (I1 and I2). The methods for calculating the RMS values of the I1 channel and I2 channel are the same.

The RMS value calculated for each current channel is stored in the unsigned 24-bit register I1RMS or I2RMS.

Figure 19-33. Current Channel RMS Calculation



The measured current channel RMS value is stored in the relevant register every 4.34 kHz. Note that, by setting the ZXRMS bit of the NLCTL register to 1, the measurement result can be stored in the register in sync with the line frequency (in sync with the zero-crossing of the input voltage). If the ZXRMS bit is set, the operation is as follows.

- If CHMD = 0:
  The system enters the two-wire mode, and the I1RMS and I2RMS registers are updated in sync with the zero-crossing of the input voltage V1 (INTZX1).

- If CHMD = 1:
  The system enters the three-wire mode, the I1RMS register is updated in sync with the zero-crossing of the input voltage V1 (INTZX1), and the I2RMS register is updated in sync with the zero-crossing of the input voltage V2 (INTZX2).

For details about INTZX1 and INTZX2, see **CHAPTER 20 POWER QUALITY MEASUREMENT CIRCUIT**.

For full-scale input, the ΔΣ-type A/D converter outputs the 23-bit conversion result 2^{20} = 0d1048576 = 0x100000. The relationship between the maximum ΔΣ-type A/D converter conversion result, the PGA (preamplifier gain), and the input voltage range is as follows:

- If PGA = 1: maximum input = 0.375 V
- If PGA = 2: maximum input = 0.1875 V
- If PGA = 16: maximum input = 0.0234375 V

The equivalent RMS (24-bit) value of a full-scale AC signal is:

$$Irms\_FS = \frac{2^{20}}{\sqrt{2}} = 0d741455 = 0xB504F$$

The current RMS measurement is accurate to within 0.5% for signal input between full scale and full scale/500.

Note that the calculated register value must be converted to amps by using the CPU.

(3) Current channel RMS offset compensation

To remove current channel offset during current channel RMS value calculation, current channel RMS offset specification registers (I1RMSOS and I2RMSOS) are provided. These are signed 12-bit registers.

One LSB of the current channel RMS offset specification register is equivalent to 2,048 LSBs of the square of the current channel RMS register value.

The offset specified for the current channel RMS offset specification register is adjusted by multiplying the offset by 2,048 and adding the square of the pre-correction RMS value. For full-scale AC input, the maximum value from the current channel RMS calculation is 0d741455. For –60 dB input (1,000: equivalent to one dynamic range), the adjustment precision of the current channel RMS offset specification register is 0.186%/LSB. For –54 dB input (500: equivalent to one dynamic range), the adjustment precision of the current channel RMS offset specification register is 0.046%/LSB.

$$Irms = \sqrt{Irms0^2 + IRMSOS \times 2048}$$

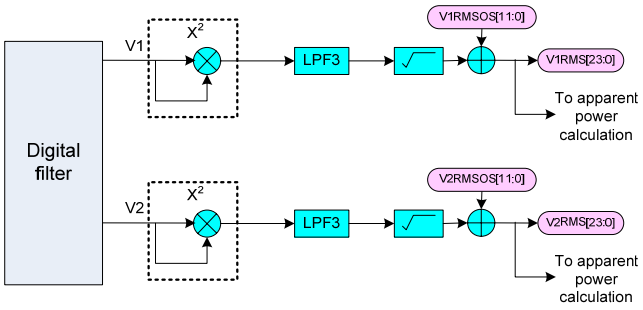
Here, Irms0 is the RMS measurement result without offset correction.

(4) Current channel RMS calculation

This circuit calculates the RMS values for the two input current channels (V1 and V2). The methods for calculating the RMS values of the V1 channel and V2 channel are the same.

The RMS value calculated for each current channel is stored in the unsigned 24-bit register V1RMS or V2RMS.

Figure 19-34. Current Channel RMS Calculation



The measured current channel RMS value is stored in the relevant register every 4.34 kHz. Note that, by setting the ZXRMS bit of the NLCTL register to 1, the measurement result can be stored in the register in sync with the line frequency (in sync with the zero-crossing of the input voltage). If the ZXRMS bit is set, the operation is as follows.

- If CHMD = 0:
  The system enters the two-wire mode, and the V1RMS and V2RMS registers are updated in sync with the zero-crossing of the input voltage V1 (INTZX1).

- If CHMD = 1:
  The system enters the three-wire mode, the V1RMS register is updated in sync with the zero-crossing of the input voltage V1 (INTZX1), and the V2RMS register is updated in sync with the zero-crossing of the input voltage V2 (INTZX2).

For details about INTZX1 and INTZX2, see **CHAPTER 20 POWER QUALITY MEASUREMENT CIRCUIT**.

The equivalent rms (24-bit) value of a full-scale AC signal is 0d741455 (0xB504F), which represents an analog input of 0.375 V x sin (ωt).

The current RMS measurement is accurate to within 0.5% for signal input between full scale and full scale/100.
Note that the calculated register value must be converted to bolts by using the CPU.

(5) Voltage channel RMS offset compensation

To remove voltage channel offset during voltage channel RMS value calculation, voltage channel RMS offset specification registers (V1RMSOS and V2RMSOS) are provided. These are signed 12-bit registers.
One LSB of the voltage channel RMS offset specification register is equivalent to 16 LSBs of the voltage channel RMS register value.
The offset specified for the voltage channel RMS offset specification register is adjusted by multiplying the offset by 16 and adding of the pre-correction RMS value. For full-scale AC input, the maximum value from the voltage channel RMS calculation is 0d741455. For –60 dB input (1,000: equivalent to one dynamic range), the adjustment precision of the voltage channel RMS offset specification register is 2.15 %/LSB. For –40 dB input (100: equivalent to one dynamic range), the adjustment precision of the voltage channel RMS offset specification register is 0.215%/LSB.

$$Vrms = Vrms0 + 16 \times VRMSOS$$

Here, Vrms0 is the RMS measurement result without offset correction.

19.4.4 Active power and energy calculation

(1) Active power calculation

(a) Active power calculation method

Active power is defined as the rate of energy flow from the power supply to the load and is the product of the voltage and current signal. The product is called the instantaneous power signal and is equal to the rate of energy flow at every instant of time. The unit of power is the watt. The following equations are used to calculate the instantaneous power signal in an AC system:

$$i(t) = \sqrt{2} \times I \sin(wt)$$

$$v(t) = \sqrt{2} \times V \sin(wt)$$

Here, V is the RMS voltage, I is the RMS current, and w is the frequency in radians.

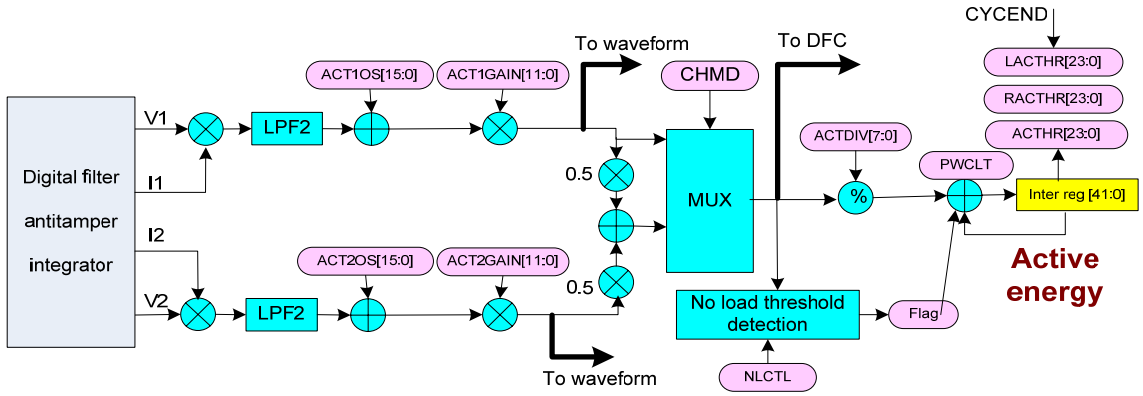
$$p(t) = v(t) \times i(t) = V \times I - V \times I \cos(2wt)$$

The average power over a specific number of line cycles (n) is calculated using the following equation:

$$P = \frac{1}{nT} \int_0^{nT} p(t)dt = V \times I$$

Here, T is the line cycle period, and P is the active or real power.
Note that the active power is equal to the DC component of the instantaneous power signal p (t), or V x I. This relationship is used to calculate the active power. The instantaneous power signal p (t) is generated by multiplying the current and voltage signals. The DC component of the instantaneous power signal is extracted by LPF2 (a low-pass filter).
This process is shown in the following figure:

Figure 19-35. Active Power Calculation



- If CHMD = 0 (the single-phase two-wire mode): active power = V1 × I1
- If CHMD = 1: (the single-phase three-wire mode): active power = 0.5 × (V1 × I1 + V2 × I2)

The low-pass filter LPF2 is used to extract the DC component of the instantaneous power, which has a frequency of f – 3 db = 5.4 Hz.

For LPF2, a certain amount of ripple occurs due to the instantaneous power signal, but this ripple is sinusoidal and has a frequency equal to twice the line frequency. For example, if the line frequency is 50 Hz, the ripple frequency is 100 Hz, and, if the line frequency is 60 Hz, the ripple frequency is 120 Hz. Because the ripple is sinusoidal, it is removed when the active power signal is integrated to calculate the energy. (For details, see **19.4.4 (2) Active energy calculation**.)

The active power signal can be read from the waveform register by setting up the SAMPMODE register and clearing the WFSMMK bit in the extended SFR (3rd SFR) interrupt mask flag register 22 (MK22). For the recommended flow, see **19.4.9 (5) Waveform-related interrupt (INTWFSM)**.

(b) Active power gain calibration

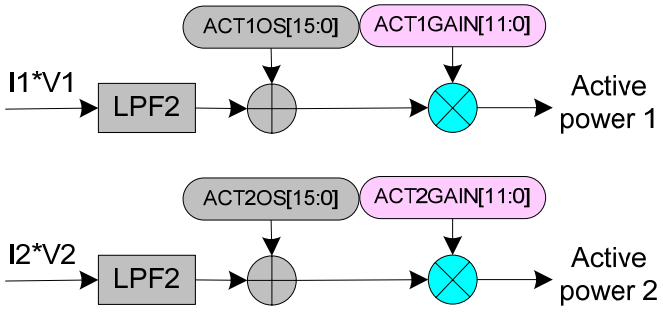
The active power is calculated by filtering the multiplier output by using a low-pass filter (LPF2). No gain calibration is performed on the current channel or voltage channel. Gain calibration is performed on active power by using the active power gain specification registers (ACT1GAIN and ACT2GAIN). The gain is adjusted by writing to the signed 12-bit registers ACT1GAIN and ACT2GAIN.

There are two active power channels in the three-wire mode:

- power1 = I1 x V1
- power2 = I2 x V2

Therefore, there are two registers called ACT1GAIN and ACT2GAIN for power1 and power2, respectively.

Figure 19-36. Active Power Gain Calibration



The following equations show how the gain is related to the ACT1GAIN and ACT2GAIN register settings:

$$OutputACT1GAIN = \{ActivePower1 \times [1 + \frac{ACT1GAIN}{2^{12}}]\}$$

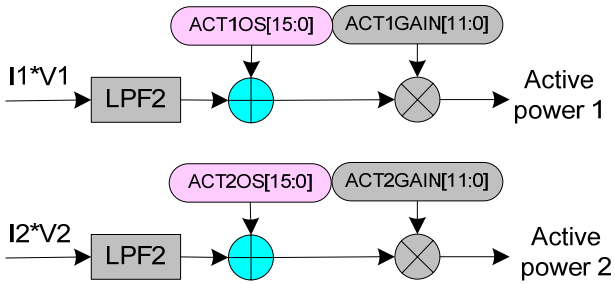
$$OutputACT2GAIN = \{ActivePower2 \times [1 + \frac{ACT2GAIN}{2^{12}}]\}$$

For example, if 0x7FF is written to the active power gain specification registers (ACT1GAIN and ACT2GAIN), the power output increases by 50% (0x7FF = 2047d, $2047/2^{12}$ = 0.5). Similarly, if 0x800 = –2048d (signed, two's complement), the power output decreases by 50%. Each LSB increases or decreases the power output by 0.0244% ($1/2^{12}$ = 0.0244%). The output range is minimized when the ACT1GAIN or ACT2GAIN register setting is equal to 0x800, and the output range is maximized by writing 0x7ff to the ACT1GAIN or ACT2GAIN register. This can be used to calibrate the active power (or energy) calculation.

(c) Active power offset calibration

The active power offset can be calibrated using the active power offset specification registers (ACT1OS and ACT2OS). These signed 16-bit registers can be used to remove offset during active power calculation. Using offset calibration keeps the active power register value at 0 when no power is being consumed.

Figure 19-37. Active Power Offset Calibration



128 LSBs (ACT1OS = 0x080 or ACT2OS = 0x080) written to the active power offset specification register are equivalent to one LSB of the LPF2 output. If the input of the voltage and current channels is at full scale , the value output by LPF2 is 0x200000 = 0d2097152. However, if the current channel input is at –60 dB (1/1000th of the full-scale current channel input), the value output by LPF2 is 0d2097.152. This means that, when a –60 dB signal is input, one LSB in the LPF2 output has a maximum measurement error of 0.0477%. One LSB of the active power offset specification register is equivalent to 1/128 LSB of LPF2 output. Therefore, when performing offset calibration, the measurement error is 0.000372%/LSB (0.0477%/128) at –60 dB.

(d) Active power sign detection

Active power sign detection detects a change of the active power sign.

The following three bits are used for this detection.

- The ACTSIGN bit in the PWCTL2 register:

  This bit selects the event that triggers an active power sign interrupt.

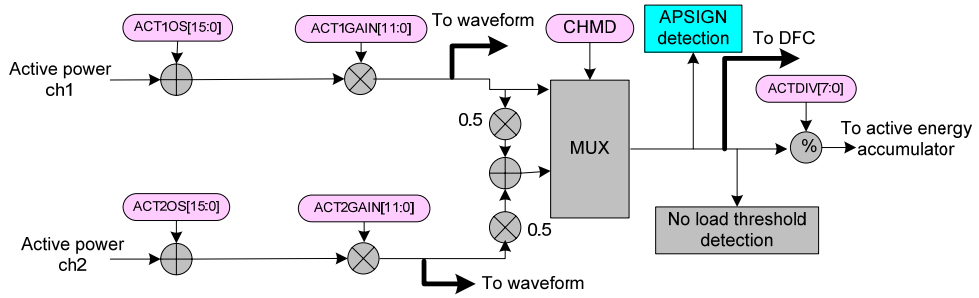
  If ACTSIGN = 0: An INTACTSIGN interrupt occurs when the active power changes from positive to negative.

  If ACTSIGN = 1: An INTACTSIGN interrupt occurs when the active power changes from negative to positive.

- The ACTSIGNIF bit in the extended SFR (3rd SFR) interrupt request flag register 20 (IF20)
  If this bit is 1, the trigger condition specified by ACTSIGN has been satisfied. Once the ACTSIGNIF interrupt status is set to 1, it remains 1 until this status bit is cleared. The ACTSIGNIF status is cleared when a zero is written to this bit.

- The ACTSIGNMK bit in the extended SFR (3rd SFR) interrupt mask flag register 20 (MK20)
  When this bit is cleared, the ACTSIGNIF flag is set and an interrupt occurs.

Figure 19-38. Active Power Offset Calibration



(e) Active-power no-load detection

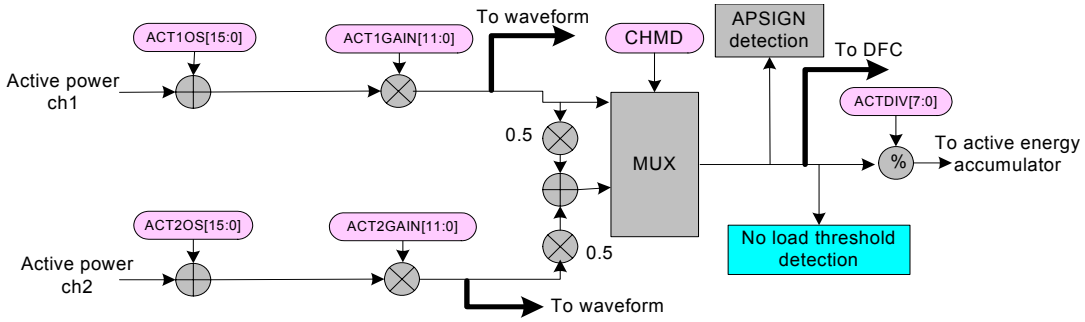
A no-load detection function is included for the active energy that eliminates any creep effects on the meter. When this function is used, active energy is not accumulated if the active power is below the no-load threshold. The following three bits are used for active-power no-load detection.

- The ACTNOLOAD1 and ACTNOLOAD0 bits in the NLCTL register:
  These two bits specify the active-power no-load threshold.
  00: Disables active-power no-load detection.
  01: Enables active-power no-load detection with a threshold of 0.015% of the full scale.
  10: Enables active-power no-load detection with a threshold of 0.0075% of the full scale.
  11: Enables active-power no-load detection with a threshold of 0.0037% of the full scale.

- The ACTNOLDIF flag in the extended SFR (3rd SFR) interrupt request flag register 21 (IF21):
  The ACTNOLDIF flag is set when the active power falls below the no-load threshold specified by the ACTNOLOAD1 and ACTNOLOAD0 bits in the NLCTL register.

- The ACTNOLDMK bit in the extended SFR (3rd SFR) interrupt mask flag register 21 (MK21):
  If the ACTNOLDMK bit is cleared in the extended SFR (3rd SFR) interrupt mask flag register 21 (MK21), an interrupt occurs. This interrupt stays active until the ACTNOLDIF status bit is cleared.

Figure 19-39. Active-power No-load Detection



(2) Active energy calculation

(a) Active energy calculation method

As stated in the active power calculation section, power is defined as the rate of energy flow. The energy can be calculated using the following equation:

$$E = \int P(t)dt$$

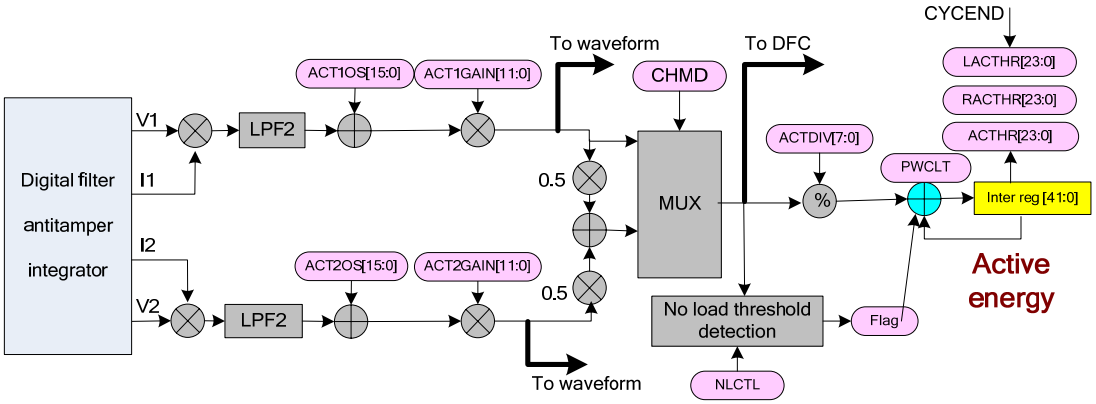
The active power signal is integrated by accumulating the active power signal in an internal 42-bit energy register. The higher 24 bits of this internal register can be read by the ACTHR register. This accumulation processing is equivalent to integration over a continuous period of time.

$$E = \int p(t)dt = \lim_{t->0}\{\sum_{n=1}^{\infty} p(nt)\times T\}$$

In the above equation, n is the discrete time sample number and T is the sample period.

The discrete time sample period (T) for the accumulation register is 230.4 μs (18 $\times$ 128/MCLK_10M). In addition to calculating the energy, this integration removes sinusoidal components from the active power signal.

Figure 19-40. Active Energy Calculation



The active energy accumulation method can be selected by specifying values for the POAM and ABSAM bits in the PWCTL2 register. For details about the accumulation method, see **19.4.4 (2) (c) Active energy accumulation modes**.

Table 19-2. Setting of Active Power Accumulation Mode

POAM	ABSAM	Active Power Accumulation Mode
0	0	Signed value accumulation mode
0	1	Absolute value accumulation mode
1	0	Positive value accumulation mode
1	1	Absolute value accumulation mode

The active power is scaled by the value of the ACTDIV register and is then accumulated in an internal energy accumulator. ACTDIV is an 8-bit unsigned register, and the scaling processing is as follows:

Amount of energy after scaling = power before scaling/ACTDIV

Note that clearing ACTDIV to 0 is prohibited. If 0 is specified, processing is performed as though 1 is specified.

After scaling, the active power is accumulated in an internal 42-bit active energy accumulator. The higher 24 bits of the internal accumulator can be read by a register. There are three methods for reading the accumulator value, and a reading register is prepared for each.

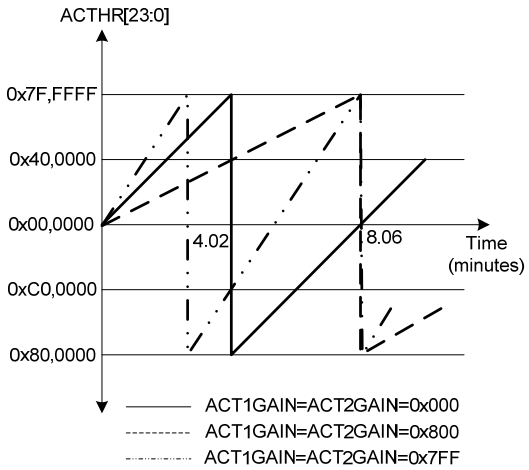
- If reading the value by using the ACTHR register:
  The value of the higher 24 bits of the accumulator at the time of reading is read.

- If reading the value by using the RACTHER register:
  The value of the higher 24 bits of the accumulator at the time of reading is read. After reading, the higher 24 bits of the accumulator are cleared.

- If reading the value by using the LACTHR register:
  The value is read in sync with the line frequency. For details, see **19.4.4 (2) (e) Line cycle active energy accumulation mode**.

The figure below shows this energy accumulation for full-scale analog-input signals (sinusoidal). The three displayed curves show the minimum period of time it takes the energy register to overflow or underflow when the active power gain specification register contents (ACT1GAIN and ACT2GAIN) are 0x7FF, 0x000, and 0x800. As shown, the fastest integration time occurs when the active power gain specification register is set to maximum full scale, which is 0x7FF.

Figure 19-41. Active Energy Accumulation for Full-scale Signals



Note that, if an overflow occurs while the power or energy flow is positive, the energy register contents invert to the maximum negative value (0x800000) and then continuously increase in value. Conversely, if the power is negative, after an underflow, the energy register contents switch to the maximum positive value (0x7FFFFF) and then continuously decrease in value.

If the active energy register is half full (positive or negative) or an overflow or underflow occurs, the interrupt flag bit ACTEHFIF or ACTEOFIF is set. By clearing the ACTEHFMK and ACTEOFMK bits in the extended SFR (3rd SFR) interrupt mask flag registers 21 and 22 (MK21 and MK22), it is possible to specify that interrupts be generated when the active energy register is half full or when an overflow occurs.

(b) Integration time under a steady load

As mentioned in the active energy calculation section, the discrete time sample period (T) for the accumulation register is 230.4 μs (4.34 kHz). If the analog input is a full-scale sinusoidal signal and the ACT1GAIN and ACT2GAIN registers are set to 0x000, the average word value from each GAIN is 2^{21} (or 0x1F, FFFF). The maximum positive value that can be stored in the internal 42-bit register before it overflows is 2^{41} (or 0x1FF, FFFF, FFFF). The integration time under these conditions when ACTDIV = 0 is calculated using the following equation:

$$Time = \frac{0x1FF,FFFF,FFFF}{0x1F,FFFF} \times 230.4us = 241.59\,\mathrm{sec} = 4.02\,\mathrm{min}$$

If ACTDIV is set to a value other than 0, the integration time varies as follows:

$$Time = Time_{ACTDIV=0} \times ACTDIV$$

(c) Active energy accumulation modes

There are three accumulation modes for active energy calculation. The mode is determined using the POAM and ABSAM bits in the PWCTL2 register.

Table 19-3. Setting of Active Power Accumulation Mode

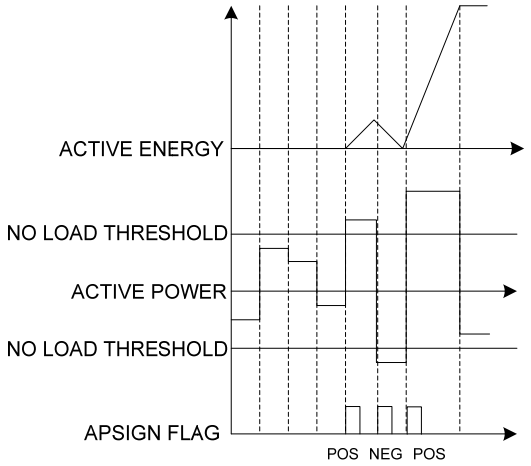
POAM	ABSAM	Active Power Accumulation Mode
0	0	Signed value accumulation mode
0	1	Absolute value accumulation mode
1	0	Positive value accumulation mode
1	1	Absolute value accumulation mode

<1> Signed accumulation mode

In the signed accumulation mode, the signed active power is accumulated. If the active power is positive, it is added to the active energy accumulator. If the active power is negative, it is subtracted from the accumulator. This mode returns to the default mode after a reset.

If the active power is below the no-load threshold, it is not accumulated.

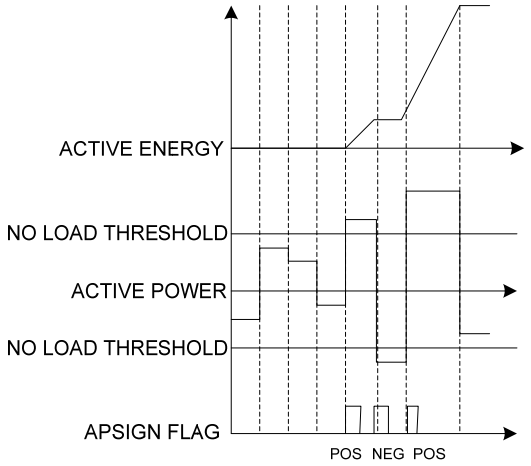
Figure 19-42. Signed Accumulation Mode (active power)



<2> Positive value accumulation mode:

This mode can be specified by setting the POAM bit in the PWCTL2 register to 1 and clearing the ABSAM bit in the same register to 0. In this mode, only positive power values are accumulated and negative power values are ignored. Negative power values and values below the no-load threshold are not accumulated.

Figure 19-43. Positive Value Accumulation Mode (Active Power)

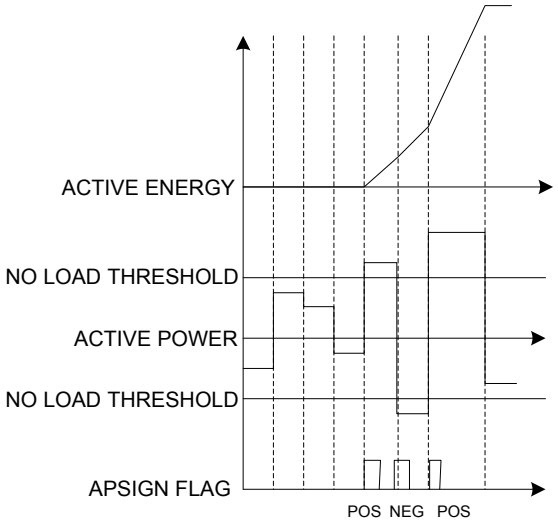


This mode also applies to CF pulse output. The default setting for this mode is off. Detection of transitions in the direction of power flow and detection of the no-load threshold are enabled in this mode.

<3> Absolute value accumulation mode:

This mode can be specified by setting the ABSAM bit in the PWCTL2 register to 1. In this mode, energy accumulation is performed using the absolute active power, ignoring active power below the no-load threshold.

Figure 19-44. Absolute Value Accumulation Mode (Active Power)

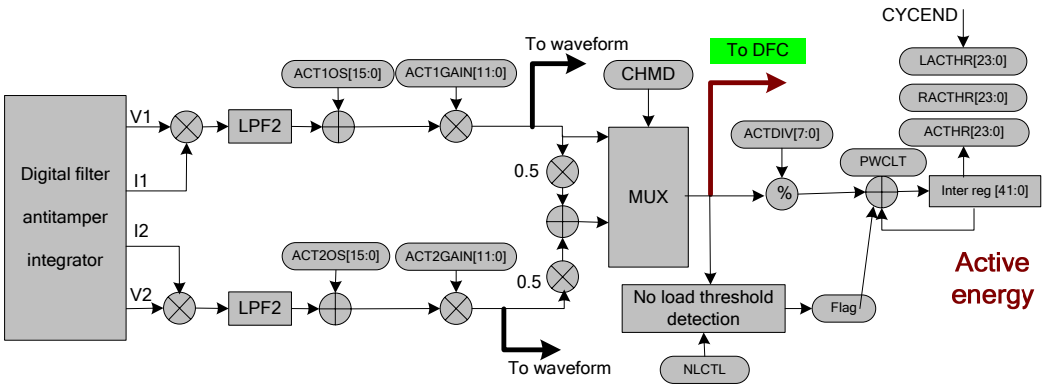


This mode also applies to CF pulse output. The default setting for this mode is off. Detection of transitions in the direction of power flow and detection of the no-load threshold are enabled in this mode.

(d) Active energy pulse output

A pulse whose frequency is proportional to the active energy is output. Outputting a pulse of this frequency uses the active power signal output by the ACT1GAIN and ACT2GAIN registers, and operation is performed in accordance with the setting of the active energy accumulation mode in the PWCTL2 register. For details about pulse output, see **CHAPTER 21 DIGITAL FREQUENCY CONVERSION CIRCUIT**.

Figure 19-45. Active Energy Pulse Output



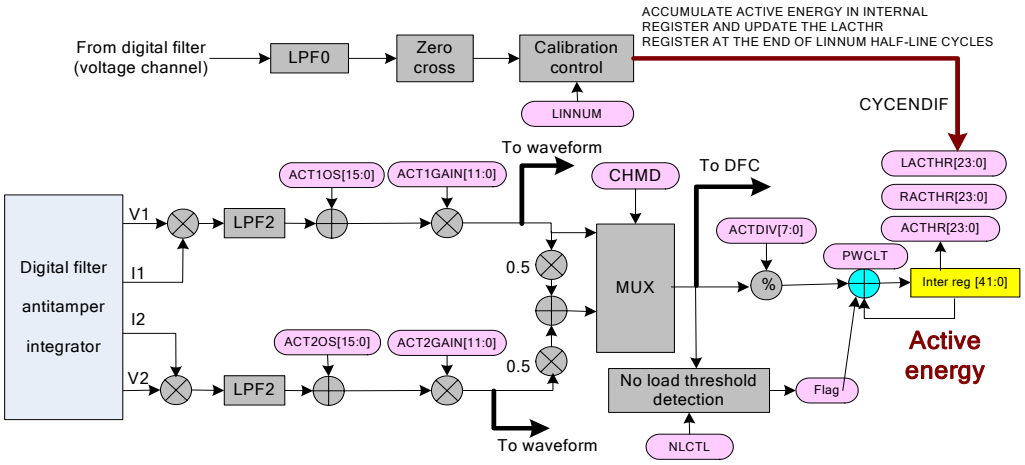
(e) Line cycle active energy accumulation mode

In the line cycle active energy accumulation mode, the error due to the sinusoidal component of the active energy is eliminated, so the active power is accumulated in units equal to twice the line frequency. This mode is implemented by synchronizing the energy accumulation with the voltage-channel zero crossing.

Active power has a ripple of 100 Hz or 120 Hz when the line frequency is 50 Hz or 60 Hz, respectively. This ripple causes calibration error when a non-integer period is accumulated during energy accumulation. This ripple has very little effect on the CF frequency because this frequency is normally very low. (For example, at 1 Hz, the accumulation time is 1 second.) However, during calibration, a short calibration time is required, and the error caused by ripple can have a large effect. If the accumulation time is equal to the integer period of the line frequency, the error caused by ripple can be eliminated. Therefore, the energy is calculated more accurately in the line cycle active energy accumulation mode.

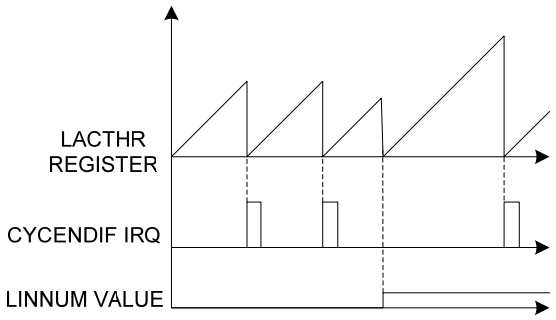
By using this mode, energy calibration can be greatly simplified, and the time necessary for calibration can be significantly reduced. In the line cycle active energy accumulation mode, the active power signal is accumulated in the LACTHR register for the specified line cycle period. The number of half-line cycles is specified in the LINNUM register, and the active energy can be accumulated for up to 65,535 half-line cycles. The active power is integrated over the specified number of line cycles, and the CYCENDIF flag in the extended SFR (3rd SFR) interrupt request flag register 22 (IF22) is set at the end of an active energy accumulation line cycle. If the CYCENDMK mask bit in the extended SFR (3rd SFR) interrupt mask flag register 22 (MK22) is cleared to 0, an interrupt occurs. This interrupt stays active until the CYCENDIF status bit is cleared. Another calibration cycle starts as soon as the CYCENDIF flag is set. If the LACTHR register is not read before a new CYCENDIF flag is set, the LACTHR register is overwritten by a new value.

Figure 19-46. Line Cycle Active Energy Accumulation Mode



When a new half-line cycle is written to the LINNUM register, the LACTHR register is reset, and a new accumulation starts at the next zero crossing. The number of half-line cycles is then counted until the LINNUM register value is reached. In this way, valid measurement is performed at the first CYCEND interrupt after writing to the LINNUM register. Line active energy accumulation uses the same signal path as active energy accumulation. The LSB size of these two registers is equivalent.

Figure 19-47. Accumulation Timing of Active Power



The CYCDIS bit in the PWCTL1 register can disable CYCEND detection. If the CYCDIS bit is set to 1, the CYCEND interrupt will not occur when the number of half-line cycles reaches the value specified for the LINNUM register. Also, if the CYCDIS bit is set to 1, the LACTHR register and inner accumulation register will not reset to 0 when a new half-line cycle is written to the LINNUM register.

Note that, in this mode, the 16-bit LINNUM register can hold a maximum value of 65,535. In other words, the line energy accumulation mode can be used to accumulate active energy for a maximum duration of 65,535 half-line cycles. At a 60 Hz line frequency, this results in a total duration of 65,535/120 Hz = 546 sec.

19.4.5 Reactive power and energy calculation

(1) Reactive power calculation

(a) Reactive power calculation method

Reactive power is defined as the product of the voltage and current waveforms when one of these signals is phase-shifted by 90°. The resulting waveform is called the instantaneous reactive power signal. The instantaneous reactive power signal when the phase of the current channel is shifted by 90° is calculated using the following equations:

$$v(t) = \sqrt{2}V\sin(\omega t + \theta) \qquad i(t) = \sqrt{2}I\sin(\omega t) \qquad i'(t) = \sqrt{2}I\sin(\omega t + \frac{\pi}{2})$$

Here, θ is the phase difference between the voltage and current channel, V is the RMS voltage, and I is the RMS current.

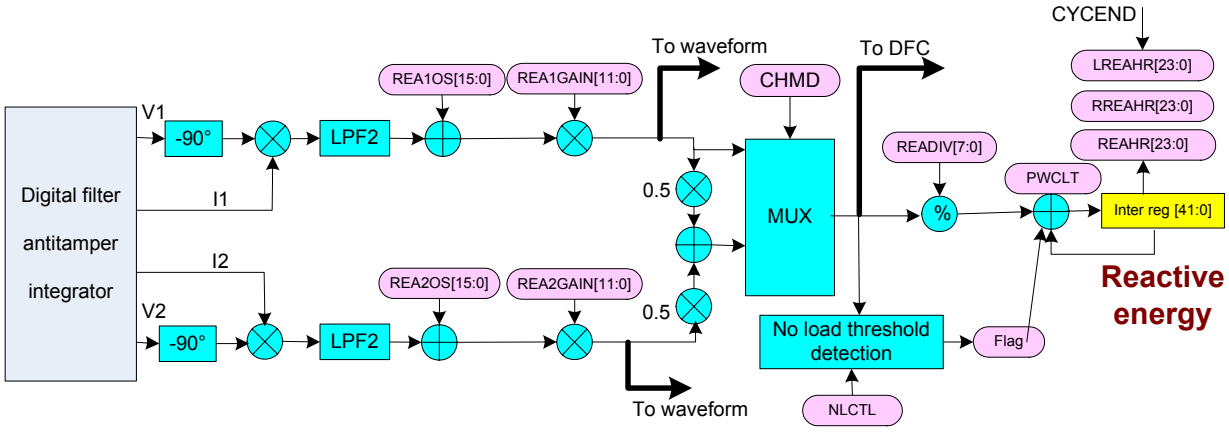
$$q(t) = V(t) \times i'(t) = VI\sin(\theta) + VI\sin(2\omega t + \theta)$$

The average reactive power for the number of line cycles (n) is calculated using the following equation:

$$Q = \frac{1}{nT}\int_0^{nT} q(t)dt = VI\sin\theta$$

Here, T is the line cycle period and Q is the reactive power.
Figure 19-48 shows the reactive power calculation process.

Figure 19-48. Reactive Power Calculation



- If CHMD = 0 (the single-phase two-wire mode): reactive power = V1_ phase-shifted × I1
- If CHMD = 1 (the single-phase three-wire mode): reactive power = 0.5 × (V1_ phase-shifted × I1 + V2_ phase-shifted × I2)

Phase shifting (a –90° phase shift) is realized using a low-pass filter (LPF) for which the high frequency component is attenuated. However, because the reactive power is calculated in the line frequency band, there are almost no harmonic effects. Gain compensation for the attenuated portion of the signal line is performed in the line frequency band. This gain compensation can be disabled using the DISREACMP bit of the NLCTL register. For details, see **19.4.2 90° phase shifter**.

The frequency response of the LPF on the reactive signal path is the same as the one used for LPF2 during average active power calculation.

For LPF2, a certain amount of ripple occurs due to the instantaneous power signal, but this ripple is sinusoidal and has a frequency equal to twice the line frequency. For example, if the line frequency is 50 Hz, the ripple frequency is 100 Hz, and, if the line frequency is 60 Hz, the ripple frequency is 120 Hz. Because the ripple is sinusoidal, it is removed when the reactive power signal is integrated to calculate the energy.

The reactive power signal can be read from the waveform register by setting up the SAMPMODE register and clearing the WFSMMK bit in the extended SFR (3rd SFR) interrupt mask flag register 22 (MK22). For the recommended flow, see **19.4.9 (5) Waveform-related interrupt (INTWFSM)**.

(b) Reactive power gain calibration

The reactive power is calculated by filtering the multiplier output by using a low-pass filter (LPF2). No gain calibration is performed on the current channel or voltage channel. Gain calibration is performed on reactive power by using the reactive power gain specification registers (REA1GAIN and REA2GAIN).

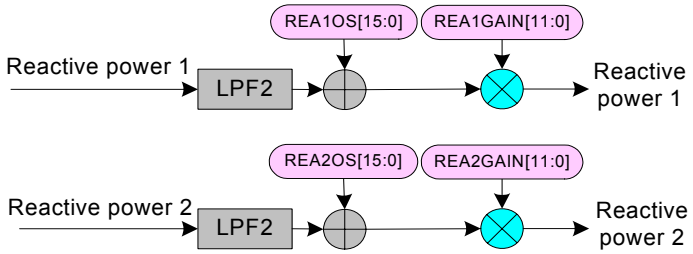
The gain is adjusted by writing to the signed 12-bit registers REA1GAIN and REA2GAIN.

There are two reactive power channels in the three-wire mode:

- power1 = I1 x V1 x sinθ1
- power2 = I2 x V2 x sinθ2

Therefore, there are two registers called REA1GAIN and REA2GAIN for power1 and power2, respectively.

Figure 19-49. Reactive Power Gain Calibration



The following equations show how the gain is related to the REA1GAIN and REA2GAIN register settings:

$$OutputREA1GAIN = \{\mathrm{Re}\, activePower1 \times [1 + \frac{REA1GAIN}{2^{12}}]\}$$

$$OutputREA2GAIN = \{\mathrm{Re}\, activePower2 \times [1 + \frac{REA2GAIN}{2^{12}}]\}$$

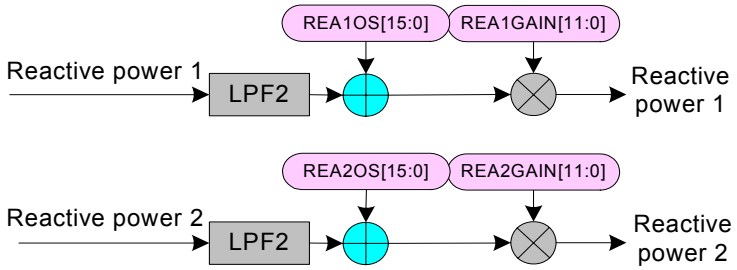
The resolution of the REA1GAIN and REA2GAIN registers is the same as the ACT1GAIN and ACT2GAIN registers (see **19.4.4 (1) (b) Active power gain calibration**).

Each LSB increases or decreases the power output by 0.0244% ($1/2^{12}$ = 0.0244%).The REA1GAIN and REA2GAIN registers can be used to adjust the reactive power (or energy) calculation.

(c) Reactive power offset calibration

The reactive power offset can be calibrated using the reactive power offset specification registers (REA1OS and REA2OS). These signed 16-bit registers can be used to remove offset during reactive power calculation. Using offset calibration keeps the reactive power register value at 0 when no power is being consumed.

Figure 19-50. Reactive Power Offset Calibration

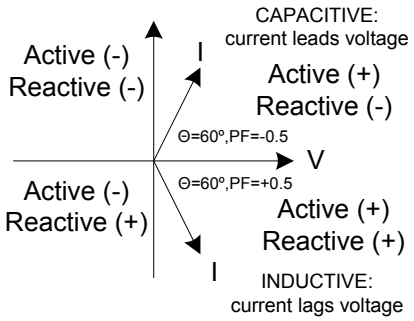


128 LSBs (REA1OS = 0x080 or REA2OS = 0x080) written to the reactive power offset specification register are equivalent to one LSB of the LPF2 output. If the input of the voltage and current channels is at full scale and the phase between the voltage and current is exactly 90 degrees, the value output by LPF2 is 0x200000 = 0d2097152. However, if the current channel input is at –60 dB (1/1000th of the full-scale current channel input), the value output by LPF2 is 0d2097.152. This means that, when a –60 dB signal is input, one LSB in the LPF2 output has a maximum measurement error of 0.0477%. One LSB of the reactive power offset specification register is equivalent to 1/128 LSB of LPF2 output. Therefore, when performing offset calibration, the measurement error is 0.000372%/LSB (0.0477%/128) at –60 dB.

(d) Reactive power sign calculation and detection

Figure 19-51 shows the signs for active power and reactive power.

Figure 19-51. Signs for Active Power and Reactive Power



Note that the average reactive power is calculated using signs. The phase shift filter performs a –90° phase shift. The following table shows the relationship of the phase difference between the voltage and the current and the sign of the reactive power calculation result.

Table 19-4. Relationship Between Phase Difference and Sign

Angle	Integrator	Sign
0° to +90°	Off	Positive
–90° to 0°	Off	Negative
0° to +90°	On	Positive
–90° to 0°	On	Negative

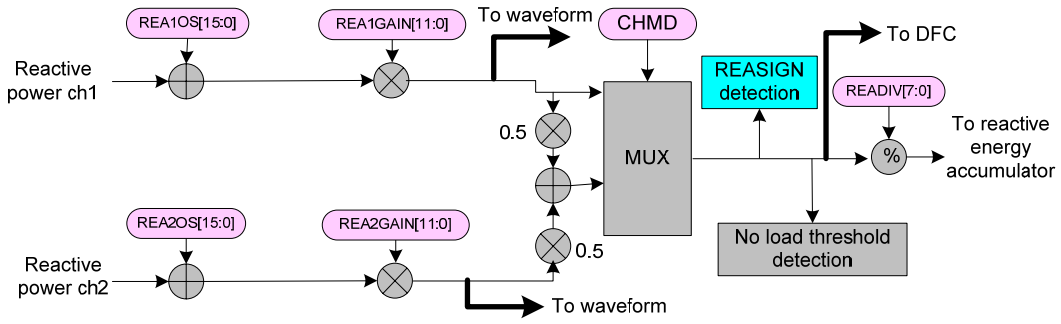
Reactive power sign detection detects a change of the active power sign.
The following three bits are used for this detection.

- The REASIGN bit in the PWCTL2 register:
  This bit selects the event that triggers an reactive power sign interrupt.
  If REASIGN = 0: An INTREASIGN interrupt occurs when the reactive power changes from positive to negative.
  If REASIGN = 1: An INTREASIGN interrupt occurs when the reactive power changes from negative to positive.

- The REASIGNIF bit in the extended SFR (3rd SFR) interrupt request flag register 20 (IF20)
  If this bit is 1, the trigger condition specified by REASIGN has been satisfied. Once the REASIGNIF interrupt status is set to 1, it remains 1 until this status bit is cleared. The REASIGNIF status is cleared when a zero is written to this bit.

- The REASIGNMK bit in the extended SFR (3rd SFR) interrupt mask flag register 20 (MK20)
  When this bit is cleared, the REASIGNIF flag is set and an interrupt occurs.

Figure 19-52. Reactive Power Offset Calibration



(e) Reactive-power no-load detection

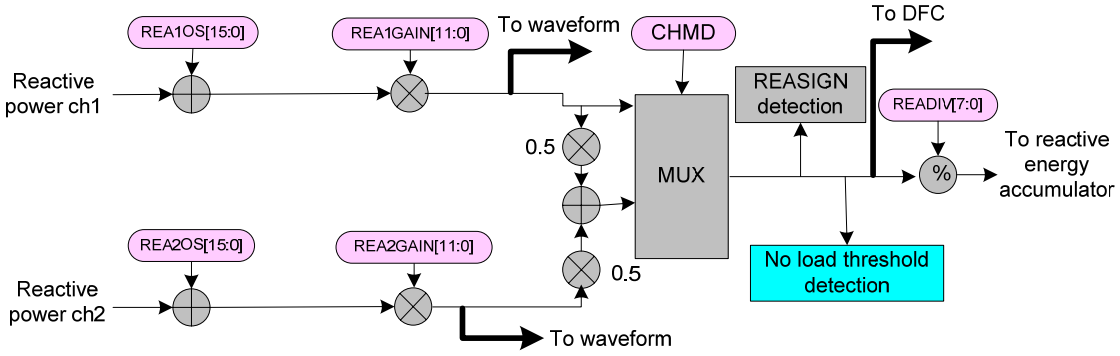
A no-load detection function is provided for the reactive power measurement. When this function is used, reactive energy is not accumulated if the reactive power is below the no-load threshold. The following three bits are used for reactive-power no-load detection.

- The REANOLOAD1 and REANOLOAD0 bits in the NLCTL register:
  These two bits specify the reactive-power no-load threshold.
  00: Disables reactive-power no-load detection.
  01: Enables reactive-power no-load detection with a threshold of 0.015% of the full scale.
  10: Enables reactive-power no-load detection with a threshold of 0.0075% of the full scale.
  11: Enables reactive-power no-load detection with a threshold of 0.0037% of the full scale.

- The REANOLDIF flag in the extended SFR (3rd SFR) interrupt request flag register 21 (IF21):
  The REANOLDIF flag is set when the reactive power falls below the no-load threshold specified by the REANOLOAD1 and REANOLOAD0 bits in the NLCTL register.

- The REANOLDMK bit in the extended SFR (3rd SFR) interrupt mask flag register 20 (MK20):
  If the REANOLDMK bit is cleared in the extended SFR (3rd SFR) interrupt mask flag register 20 (MK20), an interrupt occurs. This interrupt stays active until the REANOLDIF status bit is cleared.

Figure 19-53. Reactive-power no-load detection

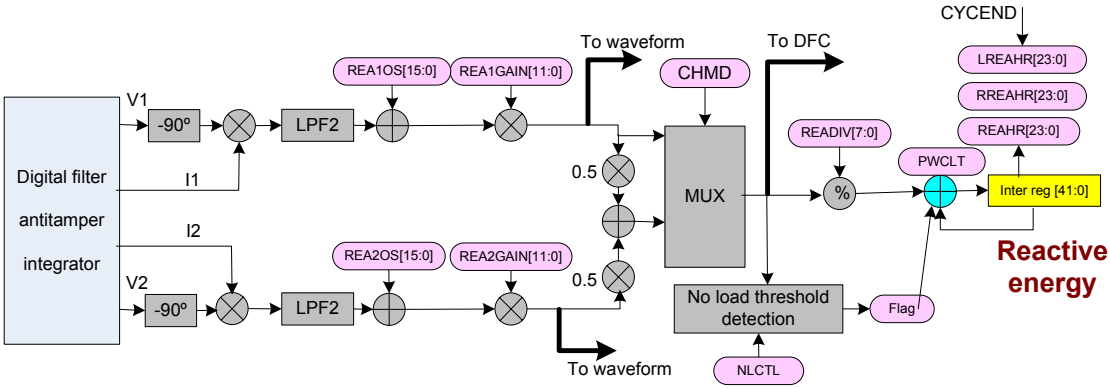


(2) Reactive energy calculation

(a) Reactive energy calculation method

As stated in the active power calculation section, reactive energy calculation is integrated by accumulating the reactive power signal in an internal 42-bit energy register. The higher 24 bits of this internal register can be read by the REAHR register. The discrete time sample period (T) for the accumulation register is 230.4 μs (4.34 kHz). In addition to calculating the energy, this integration removes sinusoidal components from the reactive power signal.

Figure 19-54. Reactive Energy Calculation



The reactive energy accumulation method can be selected by specifying values for the SAVARM and ABSVARM bits in the PWCTL2 register. For details about the accumulation method, see **19.4.5 (2) (c) Reactive energy accumulation modes**.

Table 19-5. Setting of Reactive Power Accumulation Mode

SAVARM	ABSVARM	Reactive Power Accumulation Mode
0	0	Signed value accumulation mode
0	1	Absolute value accumulation mode
1	0	Positive value accumulation mode
1	1	Absolute value accumulation mode

The reactive power is scaled by the value of the READIV register and is then accumulated in an internal energy accumulator. READIV is an 8-bit unsigned register, and the scaling processing is as follows:

Amount of energy after scaling = power before scaling/READIV

Note that clearing READIV to 0 is prohibited. If 0 is specified, processing is performed as though 1 is specified.

After scaling, the active power is accumulated in an internal 42-bit reactive energy accumulator. The higher 24 bits of the internal accumulator can be read by a register. There are three methods for reading the accumulator value, and a reading register is prepared for each.

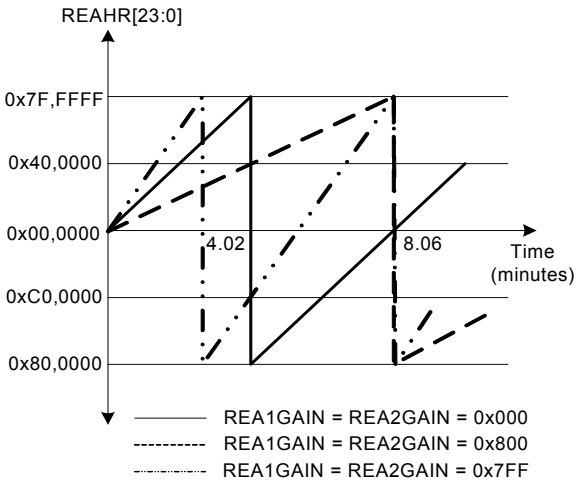
- If reading the value by using the REAHR register:
  The value of the higher 24 bits of the accumulator at the time of reading is read.

- If reading the value by using the REATHER register:
  The value of the higher 24 bits of the accumulator at the time of reading is read. After reading, the higher 24 bits of the accumulator are cleared.

- If reading the value by using the LREAHR register:
  The value is read in sync with the line frequency. For details, see **19.4.5 (2) (e) Line cycle reactive energy accumulation mode**.

The figure below shows this energy accumulation for full-scale analog-input signals (sinusoidal). The three displayed curves show the minimum period of time it takes the energy register to overflow or underflow when the reactive power gain specification register contents (REA1GAIN and REA2GAIN) are 0x7FF, 0x000, and 0x800. As shown, the fastest integration time occurs when the reactive power gain specification register is set to maximum full scale, which is 0x7FF.

Figure 19-55. Reactive Energy Accumulation for Full-scale signals



Note that, if an overflow occurs while the power or energy flow is positive, the energy register contents invert to the maximum negative value (0x800000) and then continuously increase in value. Conversely, if the power is negative, after an underflow, the energy register contents switch to the maximum positive value (0x7FFFFF) and then continuously decrease in value.

If the reactive energy register is half full (positive or negative) or an overflow or underflow occurs, the interrupt flag bit REAHFIF or REAOFIF is set. By clearing the REAHFMK and REAOFMK bits in the extended SFR (3rd SFR) interrupt mask flag register 21 (MK21), it is possible to specify that interrupts be generated when the reactive energy register is half full or when an overflow occurs.

(b) Integration time under a steady load

As mentioned in the reactive energy calculation section, the discrete time sample period (T) for the accumulation register is 230.4 μs (4.34 kHz). If the analog input is a full-scale sinusoidal signal and the REA1GAIN and REA2GAIN registers are set to 0x000, the average word value from each GAIN is 2^{21} (or 0x1F, FFFF). The maximum positive value that can be stored in the internal 42-bit register before it overflows is 2^{41} (or 0x1FF, FFFF, FFFF). The integration time under these conditions when READIV = 0 is calculated using the following equation:

$$Time = \frac{0x1FF,FFFF,FFFF}{0x1F,FFFF} \times 230.4us = 241.59\,\text{sec} = 4.02\,\text{min}$$

If READIV is set to a value other than 0, the integration time varies as follows:

$$Time = Time_{READIV=0} \times READIV$$

(c) Reactive energy accumulation modes

There are three accumulation modes for reactive energy calculation. The mode is determined using the SAVARM and ABSVARM bits in the PWCTL2 register.

Table 19-6. Setting of Reactive Power Accumulation Mode

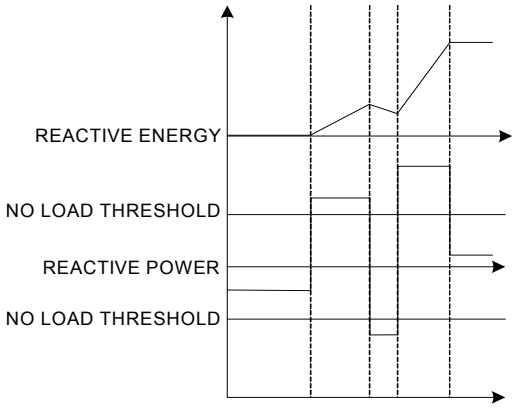
SAVARM	ABSVARM	Reactive Power Accumulation Mode
0	0	Signed value accumulation mode
0	1	Absolute value accumulation mode
1	0	Anti-tamper accumulation mode
1	1	Absolute value accumulation mode

<1> Signed accumulation mode

In the signed accumulation mode, the signed reactive power is accumulated. If the reactive power is positive, it is added to the reactive energy accumulator. If the reactive power is negative, it is subtracted from the accumulator. This mode returns to the default mode after a reset.

If the reactive power is below the no-load threshold, it is not accumulated.

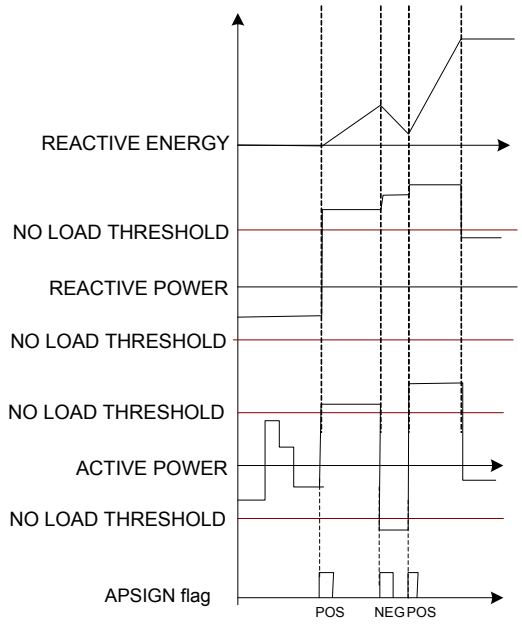
Figure 19-56. Signed Accumulation Mode (Reactive Power)



<2> Anti-tamper accumulation mode:

The anti-tamper accumulation mode can be specified by setting the SAVARM bit in the PWCTL2 register to 1. In this mode, the reactive power is accumulated in accordance with the sign of the active power. If the active power is positive, the reactive power is added as is to the reactive energy accumulator. If the active power is negative, the reactive power is subtracted from the reactive energy accumulator.

Figure 19-57. Anti-tamper Accumulation Mode (Reactive Power)

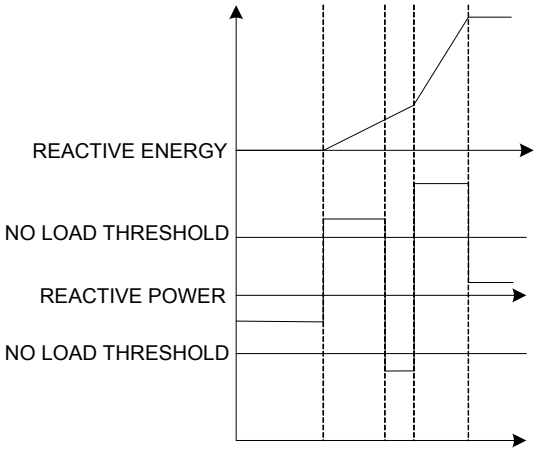


This mode also applies to CF pulse output. The default setting for this mode is off. Detection of transitions in the direction of power flow and detection of the no-load threshold are enabled in this mode.

<3> Absolute value accumulation mode:

This mode can be specified by setting the ABSVARM bit in the PWCTL2 register to 1. In this mode, energy accumulation is performed using the absolute reactive power, ignoring reactive power below the no-load threshold.

Figure 19-58. Absolute Value Accumulation Mode (Reactive Power)

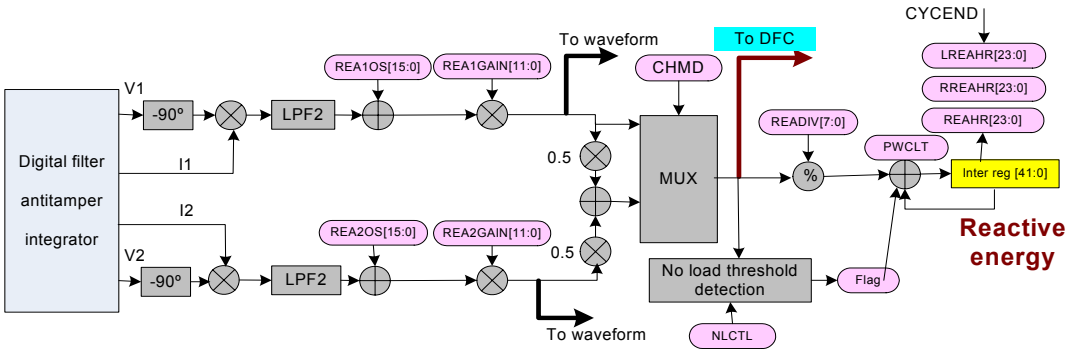


This mode also applies to CF pulse output. The default setting for this mode is off. Detection of transitions in the direction of power flow and detection of the no-load threshold are enabled in this mode.

(d) Reactive energy pulse outputs

A pulse whose frequency is proportional to the reactive energy is output. Outputting a pulse of this frequency uses the reactive power signal output by the REA1GAIN and REA2GAIN registers, and operation is performed in accordance with the setting of the reactive energy accumulation mode in the PWCTL2 register. For details about pulse output, see **CHAPTER 21 DIGITAL FREQUENCY CONVERSION CIRCUIT**.

Figure 19-59. Reactive Energy Pulse Output



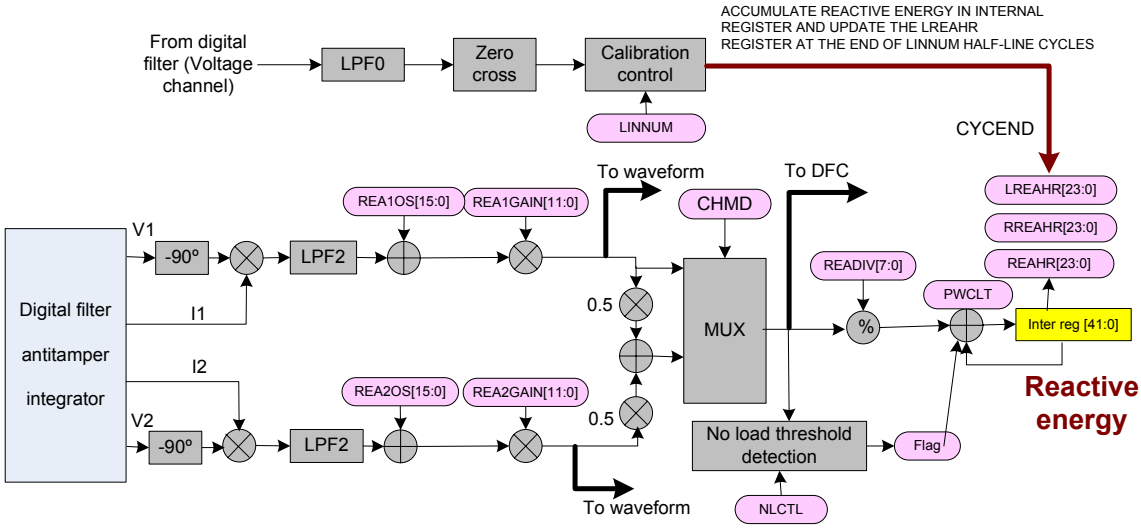
(e) Line cycle reactive energy accumulation mode

In the line cycle reactive energy accumulation mode, the error due to the sinusoidal component of the reactive energy is eliminated, so the reactive power is accumulated in units equal to twice the line frequency. This mode is implemented by synchronizing the energy accumulation with the voltage-channel zero crossing.

Reactive power has a ripple of 100 Hz or 120 Hz when the line frequency is 50 Hz or 60 Hz, respectively. This ripple causes calibration error when a non-integer period is accumulated during energy accumulation. This ripple has very little effect on the CF frequency because this frequency is normally very low. (For example, at 1 Hz, the accumulation time is 1 second.) However, during calibration, a short calibration time is required, and the error caused by ripple can have a large effect. If the accumulation time is equal to the integer period of the line frequency, the error caused by ripple can be eliminated. Therefore, the energy is calculated more accurately in the line cycle reactive energy accumulation mode.

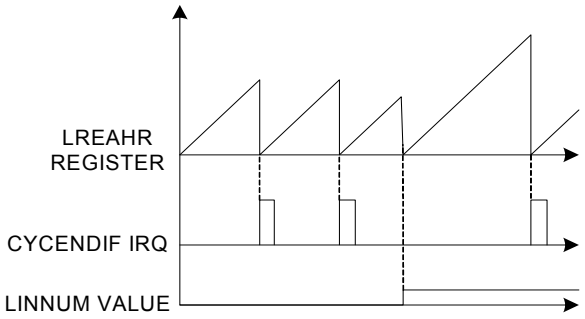
By using this mode, energy calibration can be greatly simplified, and the time necessary for calibration can be significantly reduced. In the line cycle reactive energy accumulation mode, the reactive power signal is accumulated in the LREAHR register for the specified line cycle period. The number of half-line cycles is specified in the LINNUM register, and the reactive energy can be accumulated for up to 65,535 half-line cycles. The reactive power is integrated over the specified number of line cycles, and the CYCENDIF flag in the extended SFR (3rd SFR) interrupt request flag register 22 (IF22) is set at the end of an reactive energy accumulation line cycle. If the CYCENDMK mask bit in the extended SFR (3rd SFR) interrupt mask flag register 22 (MK22) is cleared to 0, an interrupt occurs. This interrupt stays active until the CYCENDIF status bit is cleared. Another calibration cycle starts as soon as the CYCENDIF flag is set. If the LREAHR register is not read before a new CYCENDIF flag is set, the LREAHR register is overwritten by a new value.

Figure 19-60. Line Cycle Reactive Energy Accumulation Mode



When a new half-line cycle is written to the LINNUM register, the LREAHR register is reset, and a new accumulation starts at the next zero crossing. The number of half-line cycles is then counted until the LINNUM register value is reached. In this way, valid measurement is performed at the first CYCEND interrupt after writing to the LINNUM register. Line reactive energy accumulation uses the same signal path as reactive energy accumulation. The LSB size of these two registers is equivalent.

Figure 19-61. Accumulation Timing of Reactive Power



The CYCDIS bit in the PWCTL1 register can disable CYCEND detection. If the CYCDIS bit is set to 1, the CYCEND interrupt will not occur when the number of half-line cycles reaches the value specified for the LINNUM register. Also, if the CYCDIS bit is set to 1, the LREAHR register and inner accumulation register will not reset to 0 when a new half-line cycle is written to the LINNUM register.

Note that, in this mode, the 16-bit LINNUM register can hold a maximum value of 65,535. In other words, the line energy accumulation mode can be used to accumulate reactive energy for a maximum duration of 65,535 half-line cycles. At a 60 Hz line frequency, this results in a total duration of 65,535/120 Hz = 546 sec.

19.4.6 Apparent power and energy calculation

(1) Apparent power calculation

(a) Apparent power calculation method

Apparent power is defined as the maximum power that can be delivered to a load. Vrms and Irms are the effective voltage and current delivered to the load, respectively. Therefore, for the single-phase two-wire mode, the apparent power (AP) = V1RMS x I1RMS, and, for the single-phase three-wire mode, (AP) = 0.5 x (V1RMS x I1RMS + V2RMS x I2RMS). These equations are independent of the phase angle between the current and the voltage. The following equations are used to calculate the instantaneous power signal:

$$i(t) = \sqrt{2} \times Irms \sin(wt)$$

$$v(t) = \sqrt{2} \times Vrms \sin(wt + \theta)$$

$$p(t) = v(t) \times i(t) = VrmsIrms \cos(\theta) - Vrms \times Irms \cos(2wt + \theta)$$

- If CHMD = 0 (the single-phase two-wire mode):

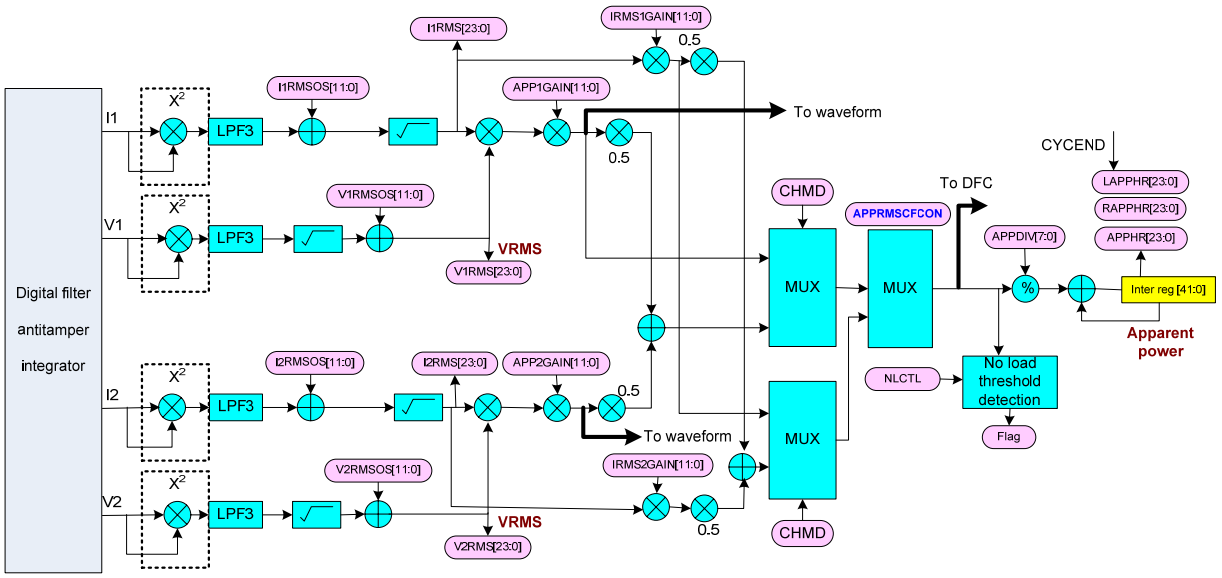
$$Apparen\_power = V1rms \times I1rms$$

- If CHMD = 1: (the single-phase three-wire mode):

$$Apparen\_power = 0.5 * (V1rms \times I1rms + V2rms \times I2rms)$$

Figure 19-62 shows the signal processing for calculating the apparent power.

Figure 19-62. Calculating the Apparent Power

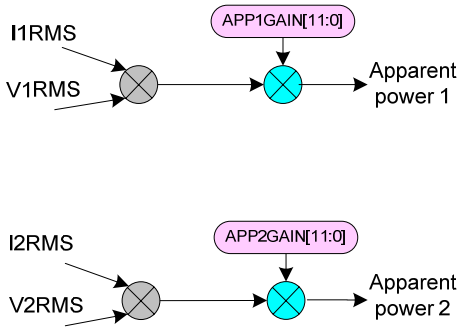


The apparent power signal can be read from the waveform register by setting up the SAMPMODE register and clearing the WFSMMK bit in the extended SFR (3rd SFR) interrupt mask flag register 22 (MK22). For the recommended flow, see **19.4.9 (5) Waveform-related interrupt (INTWFSM)**.

(b) Apparent power gain calibration

The signed 12-bit registers APP1GAIN and APP2GAIN are used to adjust the apparent energy gain.

Figure 19-63. Apparent Power Gain Calibration



The following equations show how the gain is related to the APP1GAIN and APP2GAIN register settings:

$$Output\_APP1GAIN = \{ApparentPower1 \times [1 + \frac{APP1GAIN}{2^{12}}]\}$$

$$Output\_APP2GAIN = \{ApparentPower2 \times [1 + \frac{APP2GAIN}{2^{12}}]\}$$

For example, if 0x7FF is written to the APP1GAIN register, the apparent power 1 output increases by 50% (0x7FF = 2047d, $2047/2^{12} = 0.5$). Similarly, if 0x800 = –2047d (signed, two's complement), the power output decreases by 50%. Each LSB represents 0.0244% of the power output. The apparent power is calculated using the current and voltage RMS values obtained in the RMS blocks.

(c) Apparent power offset calibration

The apparent power is the product of multiplying the average current (I1RMS, I2RMS) and average voltage (V1RMS, V2RMS). To calibrate the offset of the apparent power, use the I1RMS, I2RMS, V1RMS, and V2RMS offset specification registers. For details about offset calibration, see **19.3 (25) RMS offset specification registers for current channels 1 and 2 (I1RMSOS, I2RMSOS) and (26) RMS offset specification registers for voltage channels 1 and 2 (V1RMSOS, V2RMSOS)**.

(d) Apparent power no-load detection

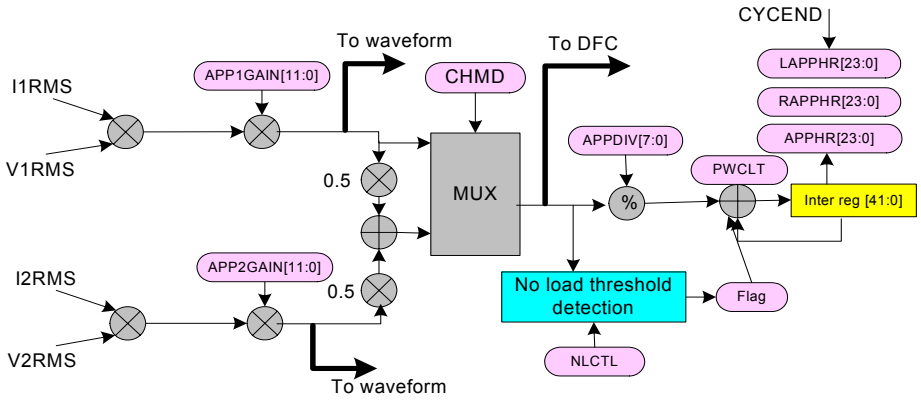
A no-load detection function is provided for the apparent power measurement. When this function is used, apparent energy is not accumulated if the apparent power is below the no-load threshold. The following three bits are used for apparent power no-load detection.

- The APPNOLOAD1 and APPNOLOAD0 bits in the NLCTL register:
  These two bits specify the apparent power no-load threshold.
  00: Disables apparent power no-load detection.
  01: Enables apparent power no-load detection with a threshold of 0.03% of the full scale.
  10: Enables apparent power no-load detection with a threshold of 0.015% of the full scale.
  11: Enables apparent power no-load detection with a threshold of 0.0075% of the full scale.

- The APPNOLDIF flag in extended SFR (3rd SFR) interrupt request flag register 21 (IF21):
  The APPNOLDIF flag is set when the apparent power falls below the no-load threshold specified by the APPNOLOAD1 and APPNOLOAD0 bits in the NLCTL register.

- The APPNOLDMK bit in the extended SFR (3rd SFR) interrupt mask flag register 20 (MK20):
  If the APPNOLDMK bit is cleared in the extended SFR (3rd SFR) interrupt mask flag register 20 (MK20), an interrupt occurs. This interrupt stays active until the APPNOLDIF status bit is cleared.

Figure 19-64. Apparent Power No-load Detection



This no-load threshold can also be applied to the CF pulse output. In this case, the no-load threshold level is the same as that for the apparent energy.

(2) Apparent energy calculation

(a) apparent energy calculation method

The apparent energy is defined as integration of apparent power.

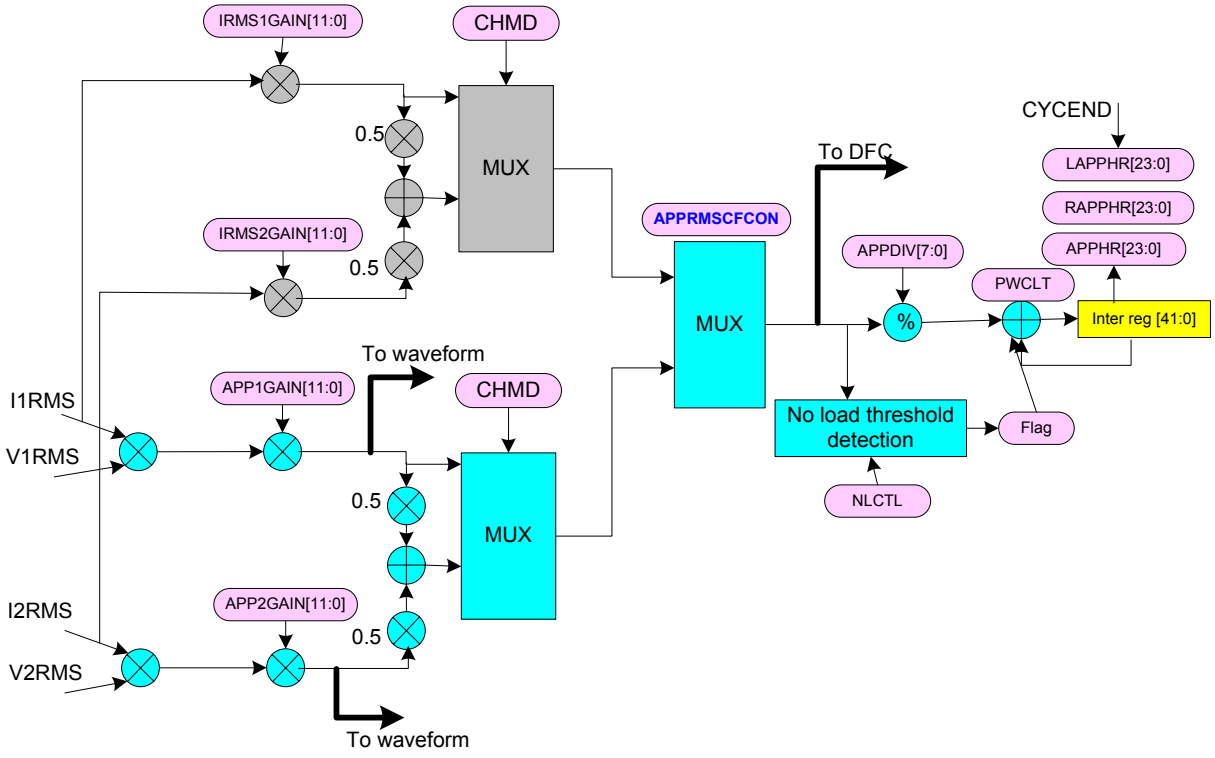
$$Apparen\_energy = \int apparent\_power(t)dt$$

The apparent power signal is integrated by accumulating the apparent power signal in an internal 42-bit energy register. The higher 24 bits of this internal register can be read by the ACTHR register.

$$Apparen\_energy = \lim_{T->0}\{\sum_{n=0}^{\infty} Apparent\_power(nT)\times T$$

In the above equation, n is the discrete time sample number and T is the sample period.
The discrete time sample period (T) for the accumulation register is 230.4 μs (4.34 kHz).

Figure 19-65. Apparent Energy Calculation



The apparent power signal is continuously added to the internal register by performing signed addition. The apparent power is scaled by the value of the APPDIV register and is then accumulated in an internal energy accumulator. APPDIV is an 8-bit unsigned register, and the scaling processing is as follows:

Amount of energy after scaling = power before scaling/APPDIV

Note that clearing APPIV to 0 is prohibited. If 0 is specified, processing is performed as though 1 is specified.

After scaling, the apparent power is accumulated in an internal 42-bit apparent energy accumulator. The higher 24 bits of the internal accumulator can be read by a register. There are three methods for reading the accumulator value, and a reading register is prepared for each.

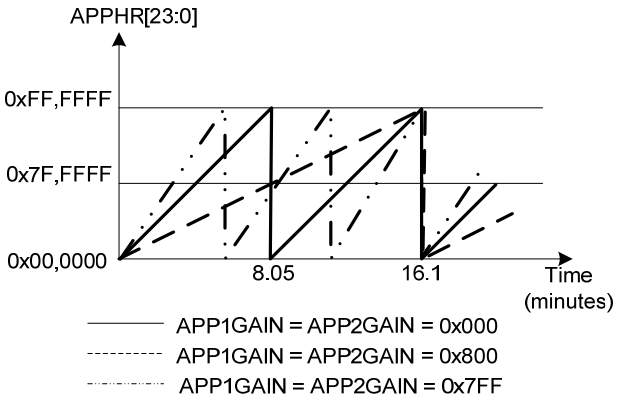
- If reading the value by using the APPHR register:
  The value of the higher 24 bits of the accumulator at the time of reading is read.

- If reading the value by using the RAPPHR register:
  The value of the higher 24 bits of the accumulator at the time of reading is read. After reading, the higher 24 bits of the accumulator are cleared.

- If reading the value by using the LAPPHR register:
  The value is read in sync with the line frequency. For details, see **19.4.6 (2) (d) Line cycle apparent energy accumulation mode**.

Because the apparent energy is always a positive value, the accumulator is always added to.

Figure 19-66. Apparent Energy Accumulation for Full-scale Signals



If the apparent energy register is half full (positive or negative) or an overflow or underflow occurs, the interrupt flag bit APPHFIF or APPOFIF is set. By clearing the APPEHFMK and APPEOFMK bits in the extended SFR (3rd SFR) interrupt mask flag register 21 (MK21), it is possible to specify that interrupts be generated when the apparent energy register is half full or when an overflow occurs. Note that, because the apparent energy register is unsigned, half-full interrupts are generated using 24 bits instead of using 23 bits as they are for the signed active energy register.

(b) Integration time under a steady load

As mentioned in the apparent energy calculation section, the discrete time sample period (T) for the accumulation register is 230.4 μs (4.34 kHz). If the analog input is a full-scale sinusoidal signal and the APP1GAIN and APP2GAIN registers are set to 0x000, the average word value from each GAIN is 2^{21} (or 0x1F, FFFF). The maximum positive value that can be stored in the internal 42-bit register before it overflows is 2^{42} (or 0x3FF, FFFF, FFFF). The integration time under these conditions when APPDIV = 0 is calculated using the following equation:

$$Time = \frac{0x3FF,FFFF,FFFF}{0x1F,FFFF} \times 230.4us = 483.13\,\text{sec} = 8.05\,\text{min}$$

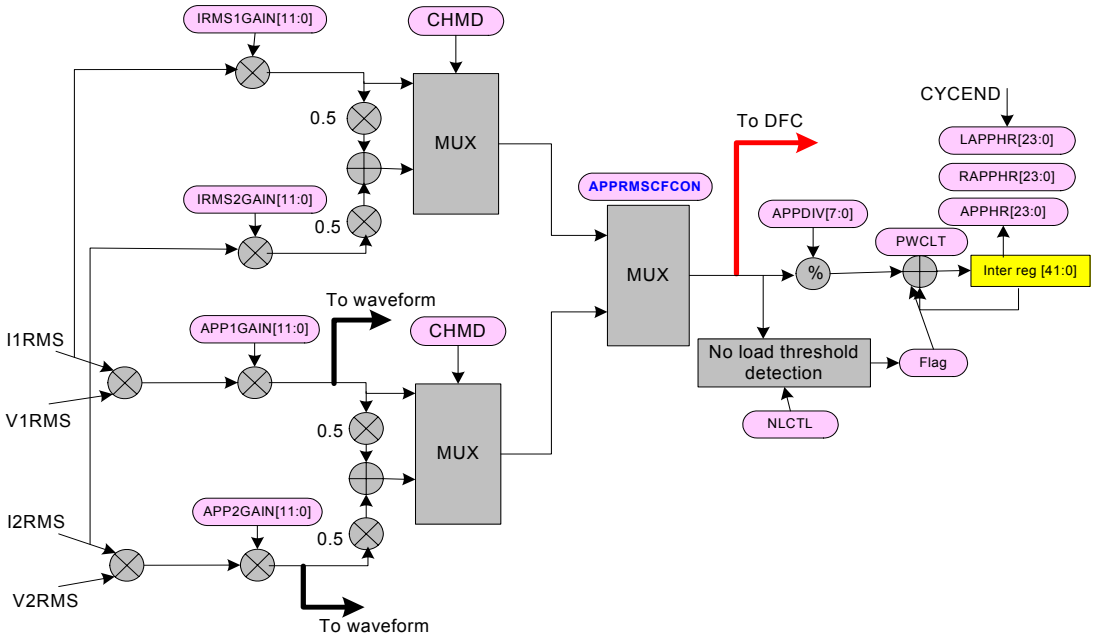
If APPDIV is set to a value other than 0, the integration time varies as follows:

$$Time = Time_{APPDIV=0} \times APPDIV$$

(c) Apparent energy pulse output

A pulse whose frequency is proportional to the apparent energy is output. Outputting a pulse of this frequency uses the apparent energy signal output by the APP1GAIN and APP2GAIN registers. This output can also be used to output a pulse whose frequency is proportional to I1rms or I2rms. For details about pulse output, see **CHAPTER 21 DIGITAL FREQUENCY CONVERSION CIRCUIT**.

Figure 19-67. Apparent Energy Pulse Output



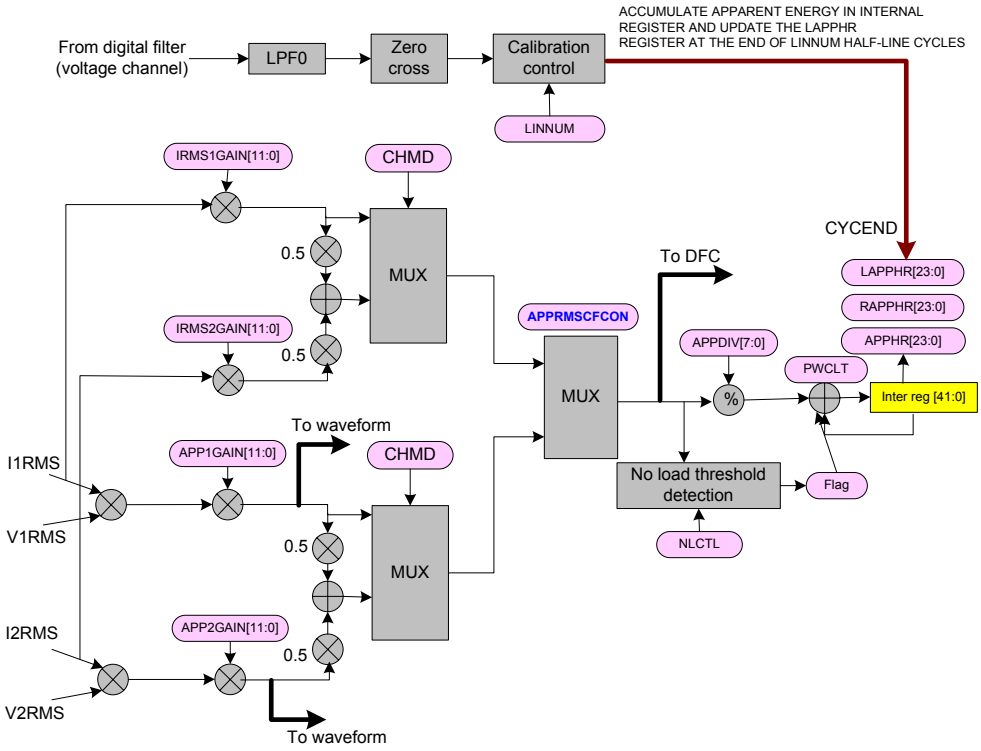
(d) Line cycle apparent energy accumulation mode

In the line cycle apparent energy accumulation mode, the error due to the sinusoidal component of the apparent energy is eliminated, so the apparent power is accumulated in units equal to twice the line frequency. This mode is implemented by synchronizing the energy accumulation with the voltage-channel zero crossing.

Apparent power has a ripple of 100 Hz or 120 Hz when the line frequency is 50 Hz or 60 Hz, respectively. This ripple causes calibration error when a non-integer period is accumulated during energy accumulation. This ripple has very little effect on the CF frequency because this frequency is normally very low. (For example, at 1 Hz, the accumulation time is 1 second.) However, during calibration, a short calibration time is required, and the error caused by ripple can have a large effect. If the accumulation time is equal to the integer period of the line frequency, the error caused by ripple can be eliminated. Therefore, the energy is calculated more accurately in the line cycle apparent energy accumulation mode.

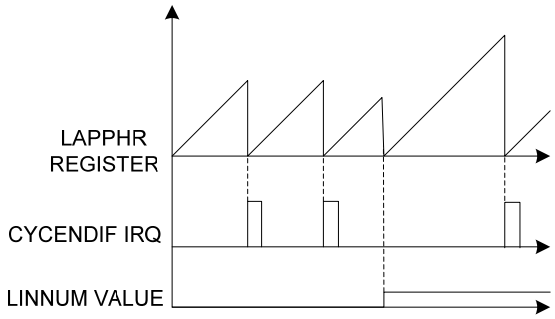
By using this mode, energy calibration can be greatly simplified, and the time necessary for calibration can be significantly reduced. In the line cycle apparent energy accumulation mode, the reactive power signal is accumulated in the LAPPHR register for the specified line cycle period. The number of half-line cycles is specified in the LINNUM register, and the apparent energy can be accumulated for up to 65,535 half-line cycles. The apparent power is integrated over the specified number of line cycles, and the CYCENDIF flag in the extended SFR (3rd SFR) interrupt request flag register 22 (IF22) is set at the end of an apparent energy accumulation line cycle. If the CYCENDMK mask bit in the extended SFR (3rd SFR) interrupt mask flag register 22 (MK22) is cleared to 0, an interrupt occurs. This interrupt stays active until the CYCENDIF status bit is cleared. Another calibration cycle starts as soon as the CYCENDIF flag is set. If the LAPPHR register is not read before a new CYCENDIF flag is set, the LAPPHR register is overwritten by a new value.

Figure 19-68. Line Cycle Apparent Energy Accumulation Mode



When a new half-line cycle is written to the LINNUM register, the LAPPHR register is reset, and a new accumulation starts at the next zero-crossing. The number of half-line cycles is then counted until the LINNUM register value is reached. In this way, valid measurement is performed at the first CYCEND interrupt after writing to the LINNUM register. Line apparent energy accumulation uses the same signal path as apparent energy accumulation. The LSB size of these two registers is equivalent.

Figure 19-69. Accumulation Timing of Apparent Power



The CYCDIS bit in the PWCTL1 register can disable CYCEND detection. If the CYCDIS bit is set to 1, the CYCEND interrupt will not occur when the number of half-line cycles reaches the value specified for the LINNUM register. Also, if the CYCDIS bit is set to 1, the LAPPHR register and inner accumulation register will not reset to 0 when a new half-line cycle is written to the LINNUM register.

Note that, in this mode, the 16-bit LINNUM register can hold a maximum value of 65,535. In other words, the line energy accumulation mode can be used to accumulate apparent energy for a maximum duration of 65,535 half-line cycles. At a 60 Hz line frequency, this results in a total duration of 65,535/120 Hz = 546 sec.

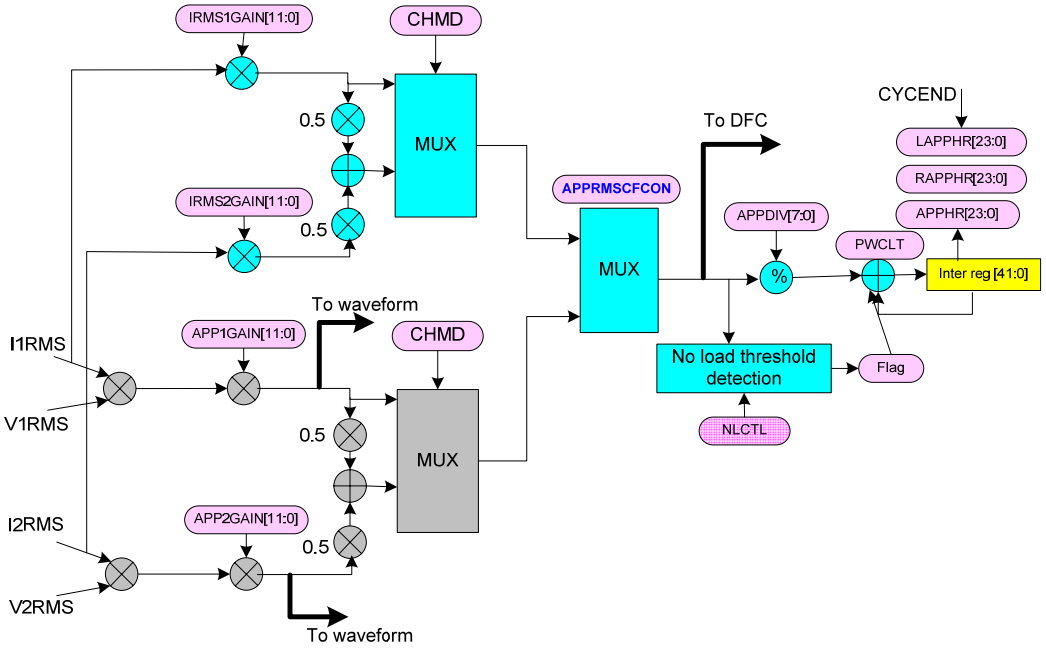
19.4.7 Ampere-hour accumulation

(1) Ampere-hour accumulation selection

Ampere-hour values can be accumulated instead of apparent power. The results of ampere-hour measurement are read using the APPHR, RAPPHR, and LAPPHR registers. Ampere-hour accumulation can be specified by setting the APPRMSCFCON bit to 1. If such accumulation is specified, digital frequency conversion also uses I1rms or 0.5 x (I1rms + I2rms) instead of the apparent power.

Note that, because apparent power accumulation and ampere-hour accumulation use the same circuits, select and use one.

Figure 19-70. Ampere-hour Accumulation



- If CHMD = 0 (the single-phase two-wire mode) : IRMS_accumulation = I1RMS
- If CHMD = 1 (the single-phase three-wire mode) : IRMS_accumulation = 0.5 x (I1RMS + I2RMS)

In this case, the IRMS_accumulation signal is used for the following accumulation and digital frequency conversion.

No-load threshold detection can be enabled by setting the APPRMSCFCON bit to 1. Specify the threshold level by using the APPNOLOAD1 and APPNOLOAD0 bits.

(2) Integration time under a steady load

The discrete time sample period (T) for the accumulation register is 230.4 μs (4.34 kHz). If the analog input is a full-scale sinusoidal signal and the RMS1GAIN and RMS2GAIN registers are set to 0x000, the average value of Irms is:

$$Irms\_FS = \frac{2^{20}}{\sqrt{2}} = 0d741455 = 0xB504F$$

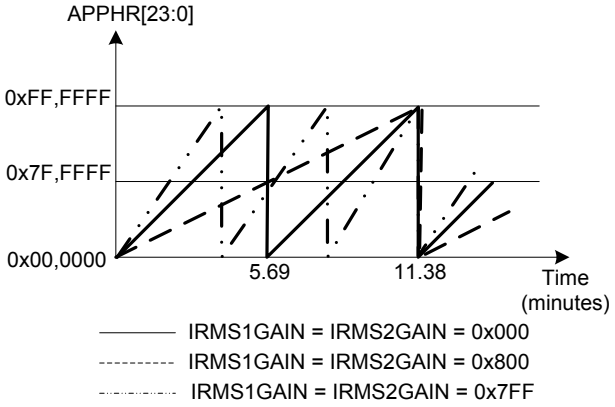
Before IRMS is accumulated in the energy accumulator, 2 LSBs of "0" are added to IRMS.

$$Irms\_acc\_FS = \frac{2^{22}}{\sqrt{2}} = 0d2965820 = 0x2D413C$$

IRMS accumulation is unsigned just like apparent power accumulation. The maximum positive value that can be stored in the internal 42-bit register is 2^{42} (or 0x3FF, FFFF, FFFF). The integration time under these conditions when APPDIV = 0 is calculated using the following equation:

$$Time = \frac{0x3FF,FFFF,FFFF}{0x2D413C} \times 230.4us = 341.66\sec = 5.69\min$$

Figure 19-71. Full-Scale Signal Ampere-hour Accumulation



When APPDIV is set to a value other than 0, the integration time varies, as shown by the following equation:

$$Time = Time_{APPDIV=0} \times APPDIV$$

19.4.8 Waveform sampling function

There are two registers (SAMP1 and SAMP2) used to read the intermediate data for power calculation. The data to read is selected using the SAMPMODE register, and up to two signals can be read at the same time. The specified data is read at a rate of 4.34 kHz and is latched to the SAMP1 and SAMP2 registers.

Figure 19-72. Waveform Sampling Function

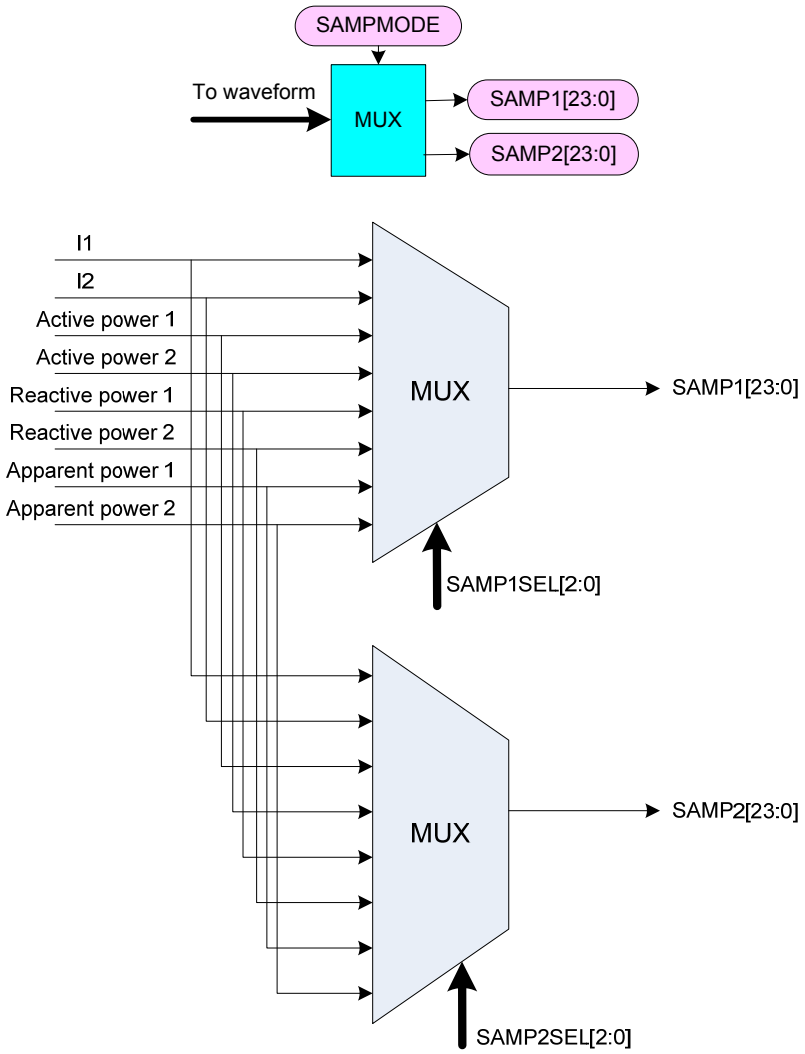


Table 19-7. Select Waveform for Sample Mode by SAMPMODE Register

SAMPnSEL2	SAMPnSEL1	SAMPnSEL0	Select the waveform n for sample mode by SAMPMODE register (n = 1, 2)	Fs_code
0	0	0	I1 output from integrator	2^{20}
0	0	1	I2 output from integrator	2^{20}
0	1	0	Active power 1 (output of ACT1GAIN after multiplication)	2^{21}
0	1	1	Active power 2 (output of ACT2GAIN after multiplication)	2^{21}
1	0	0	Reactive power 1 (output of REA1GAIN after multiplication)	2^{21}
1	0	1	Reactive power 2 (output of REA2GAIN after multiplication)	2^{21}
1	1	0	Apparent power 1 (output of APP1GAIN after multiplication)	2^{21}
1	1	1	Apparent power 2 (output of APP2GAIN after multiplication)	2^{21}

19.4.9 Interrupt

There are 13 interrupts in the power calculation circuit.

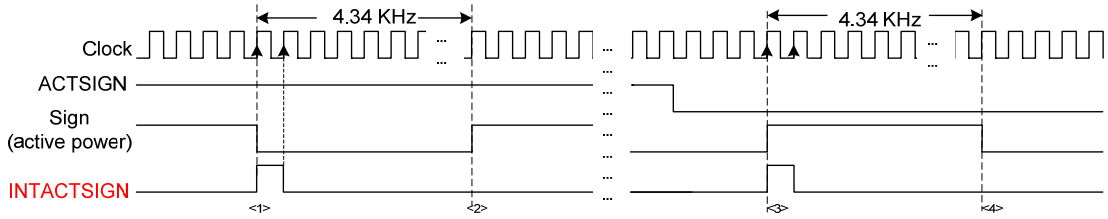
(1) Active power and energy-related interrupts

(a) INTACTSIGN

If ACTSIGN = 0, an INTACTSIGN interrupt occurs when the active power changes from positive to negative.

If ACTSIGN = 1, an INTACTSIGN interrupt occurs when the active power changes from negative to positive.

Figure 19-73. INTACTSIGN Interrupt (Active Power)



<1> If ACTSIGN = 1 and the active power changes from negative to positive, an INTACTSIGN interrupt occurs.

<2> If ACTSIGN = 1 and the active power changes from positive to negative, no INTACTSIGN interrupt occurs.

<3> If ACTSIGN = 0 and the active power changes from positive to negative, an INTACTSIGN interrupt occurs.

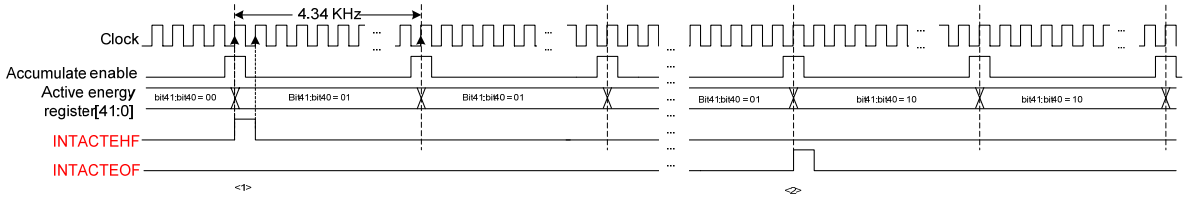
<4> If ACTSIGN = 0 and the active power changes from negative to positive, no INTACTSIGN interrupt occurs.

(b) INTACTEHF, INTACTEOF

An INTACTEHF interrupt occurs when the active energy register is half full (positive or negative).

An INTACTEOF interrupt occurs when the active energy register overflows or underflows (positive or negative).

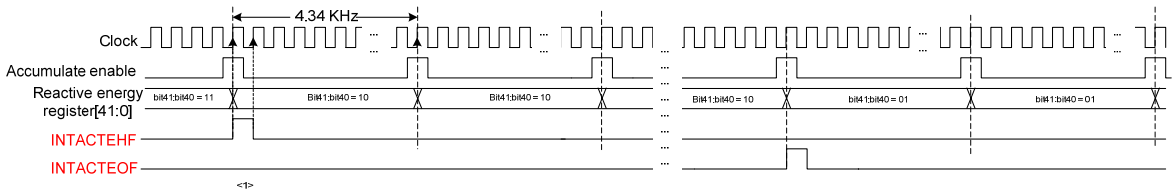
Figure 19-74. INTACTEHF and INTACTEOF Interrupts (When Active Energy Is Positive)



<1> If the active energy is half full, an INTACTEHF interrupt occurs.

<2> If the active energy overflows, an INTACTEOF interrupt occurs.

Figure 19-75. INTACTEHF and INTACTEOF Interrupts (When Active Energy Is Negative)



<1> If the active energy is half full, an INTACTEHF interrupt occurs.

<2> If the active energy overflows, an INTACTEOF interrupt occurs.

(c) INTACTNOLD

An INTACTNOLD interrupt occurs when the active power falls below the no-load threshold.

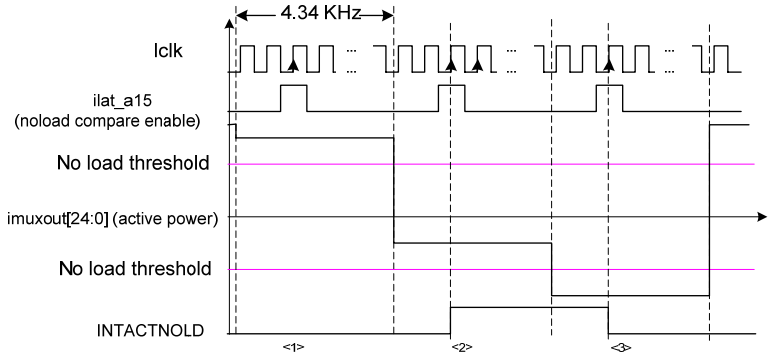
The threshold is specified by using the ACTNOLOAD1 and ACTNOLOAD0 bits in the NLCTL register. (For details, see **19.3 (3) No-load level control register (NLCTL)**).

The full-scale active power is 2^{21} = 1FFFFFH. Table 19-8 shows the threshold values for each combination of bit settings.

Table 19-8. No-load Threshold of Active Power

ACTNOLOAD1, 0	No-load threshold
00	0
01	25'h013A (0.015% of the full scale)
10	25'h009D (0.0075% of the full scale)
11	25'h004D (0.0037% of the full scale)

Figure 19-76. INTACTNOLD Interrupt (Active Power)



<1> If the active power rises above the no-load threshold, no INTACTNOLD interrupt occurs.

<2> If the active power falls below the no-load threshold, an INTACTNOLD interrupt occurs.

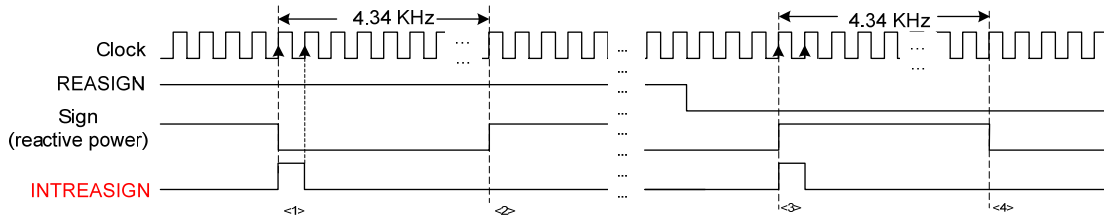
<3> If the absolute value of the active power rises above the no-load threshold, no INTACTNOLD interrupt occurs.

(2) Reactive power and energy-related interrupts

(a) INTREASIGN

If REASIGN = 0, an INTREASIGN interrupt occurs when the reactive power changes from positive to negative.
If REASIGN = 1, an INTREASIGN interrupt occurs when the reactive power changes from negative to positive.

Figure 19-77. INTREASIGN Interrupt (Reactive Power)

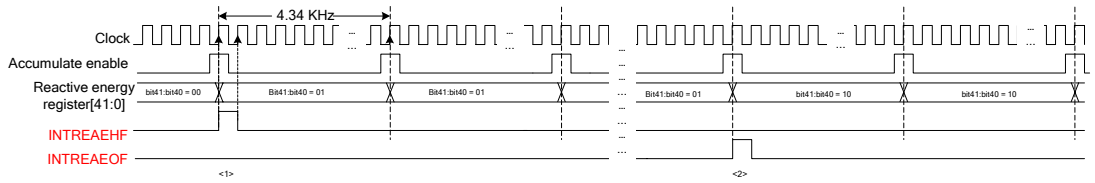


<1> If REASIGN = 1 and the reactive power changes from negative to positive, an INTREASIGN interrupt occurs.
<2> If REASIGN = 1 and the reactive power changes from positive to negative, no INTREASIGN interrupt occurs.
<3> If REASIGN = 0 and the reactive power changes from positive to negative, an INTREASIGN interrupt occurs.
<4> If REASIGN = 0 and the reactive power changes from negative to positive, no INTREASIGN interrupt occurs.

(b) INTREAEHF, INTREAEOF

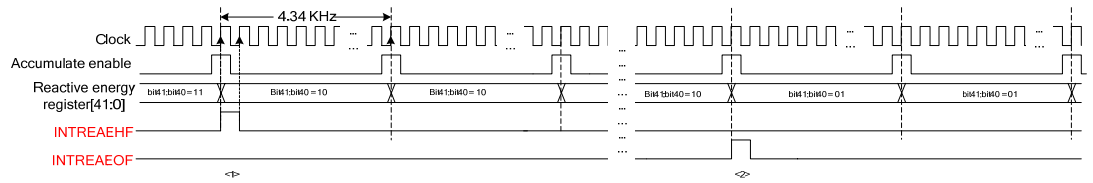
An INTREAEHF interrupt occurs when the reactive energy register is half full (positive or negative).
An INTREAEOF interrupt occurs when the reactive energy register overflows or underflows (positive or negative).

Figure 19-78. INTREAEHF and INTREAEOF Interrupts (When Reactive Energy Is Positive)



<1> If the reactive energy is half full, an INTREAEHF interrupt occurs.
<2> If the reactive energy overflows, an INTREAEOF interrupt occurs.

Figure 19-79. INTREAEHF and INTREAEOF Interrupts (When Reactive Energy Is Negative)



<1> If the reactive energy is half full, an INTREAEHF interrupt occurs.
<2> If the reactive energy overflows, an INTREAEOF interrupt occurs.

(c) INTREANOLD

An INTREANOLD interrupt occurs when the reactive power falls below the no-load threshold.

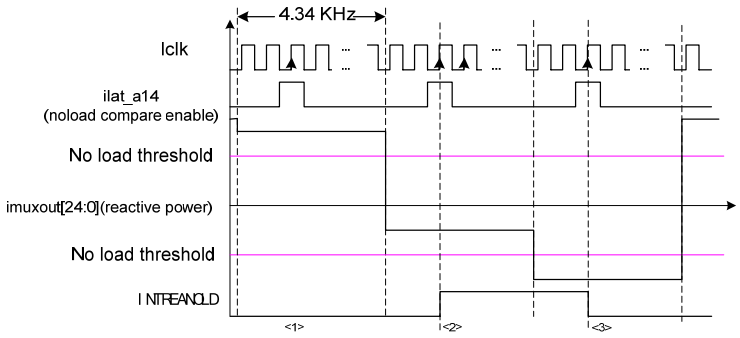
The threshold is specified by using the REANOLOAD1 and REANOLOAD0 bits in the NLCTL register. (For details, see **19.3 (3) No-load level control register (NLCTL)**.)

The full-scale reactive power is 2^{21} = 1FFFFFH. Table 19-9 shows the threshold values for each combination of bit settings.

Table 19-9. No-load Threshold of Reactive Power

REANOLOAD1, 0	No-load threshold
00	0
01	25'h013A (0.015% of the full scale)
10	25'h009D (0.0075% of the full scale)
11	25'h004D (0.0037% of the full scale)

Figure 19-80. INTREANOLD Interrupt (Reactive Power)



<1> If the reactive power rises above the no-load threshold, no INTREANOLD interrupt occurs.

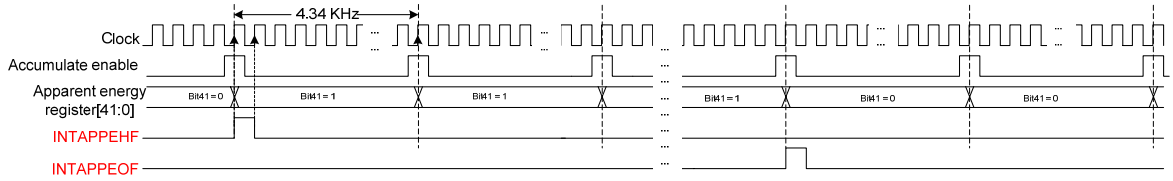
<2> If the reactive power falls below the no-load threshold, an INTREANOLD interrupt occurs.

<3> If the absolute value of the reactive power rises above the no-load threshold, no INTREANOLD interrupt occurs.

(3) Apparent power and energy-related interrupts

(a) INTAPPEHF, INTAPPEOF

Figure 19-81. INTAPPEHF and INTAPPEOF Interrupts



(b) INTAPPNOLD

An INTAPPNOLD interrupt occurs when the apparent power falls below the no-load threshold.

The threshold is specified by using the APPNOLOAD1 and APPNOLOAD0 bits in the NLCTL register. (For details, see **19.3 (3) No-load level control register (NLCTL)**.)

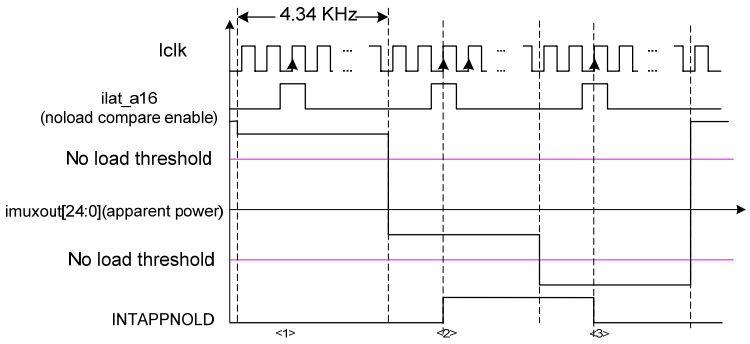
The full-scale apparent power is 2^{21} = 1FFFFFH. Table 19-10 shows the threshold values for each combination of bit settings.

Note that this no-load threshold also applies to the Irms pulse output if the APPRMSCFCON bit is set. In this case, if APPRMSCFCON = 1, IRMS no-load threshold detection is enabled, and the level of the no-load threshold is the same as for the apparent energy. The full-scale IRMS value is $2^{19.5}$. This is because, due to the fact that IRMS is 23 bits and the power is 25 bits, IRMS is expanded by two bits. The IRMS threshold is $2^{(19.5+2)} = 2^{21.5}$ = 2D413D H.

Table 19-10. No-load Threshold of Apparent Power

APPRMSCFCON	APPNOLOAD1, 0	No-load threshold
0	00	0
0	01	25'h0275 (0.03% of the apparent power full scale)
0	10	25'h013A (0.015% of the apparent power full scale)
0	11	25'h009D (0.0075% of the apparent power full scale)
1	00	0
1	01	25'h0379 (0.03% of the IRMS full scale)
1	10	25'h01BC (0.015% of the IRMS full scale)
1	11	25'h00DE (0.0075% of the IRMS full scale)

Figure 19-82. INTAPPNOLD Interrupt

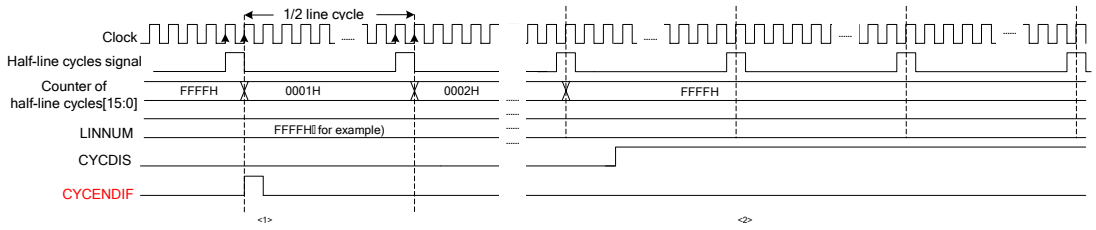


<1> If the apparent power rises above the no-load threshold, no INTAPPNOLD interrupt occurs.
<2> If the apparent power falls below the no-load threshold, an INTAPPNOLD interrupt occurs.
<3> If the absolute value of the apparent power rises above the no-load threshold, no INTAPPNOLD interrupt occurs.

(4) Cycend-related interrupt (INTCYCED)

An INTCYCED interrupt occurs when the number of half-line cycles becomes equal to the LINNUM register value.
The CYCDIS bit in the PWCTL1 register can be used to disable INTCYCED detection. If the CYCDIS bit is set to 1, no INTCYCED interrupt occurs when the number of half-line cycles reaches the value specified for the LINNUM register.

Figure 19-83. INTCYCED Interrupt



<1> If CYCDIS = 0 and the number of half-line cycles = the LINNUM value, an INTCYCED interrupt occurs.
<2> If CYCDIS = 1 and the number of half-line cycles = the LINNUM value, no INTCYCED interrupt occurs.

(5) Waveform-related interrupt (INTWFSM)

INTWFSM interrupts occur according to the settings for SAMPnSEL2, SAMPnSEL1, and SAMPnSEL0 (For details, see **19.3 (27) Sampling mode selection register (SAMPMODE)**).
When the signal assigned to SAMPnSEL changes, the corresponding interrupt occurs.

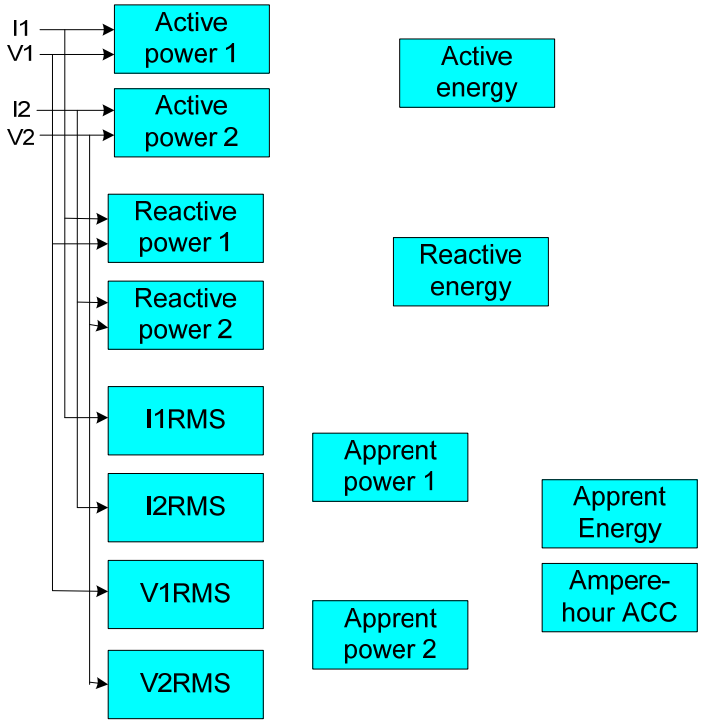
The SAMP1 and SAMP2 registers are updated during each sampling period (fs = 4.34 kHz). These registers can be read at any time. The following procedure is recommended for waveform sampling:

<1> Set up the SAMPMODE register to select the waveform source for the SAMP1 and SAMP2 registers.
<2> Set the SAMPEN bit to enable waveform sampling. Clear the WFSMMK bit of the extended SFR (3rd SFR) interrupt mask flag register 22 (MK22) to enable interrupts to occur after updating the SAMP1 and SAMP2 registers.
<3> If an INTWFSM interrupt occurs, read the SAMP1 and SAMP2 registers to obtain new waveform sampling data, and then clear WFSMIF (the interrupt flag for waveform sampling).

19.4.10 Power-saving mode

A power-saving mode is available to reduce power consumption. This mode can stop unnecessary power calculations.

Figure 19-84. Power Configuration



There are 6 bits those used to control power saving function.

- PWREN (Bit 7 of PWCTL1 register)
  0: All Power calculation functions are disable
  1: All Power calculation functions are enable

- CHMD (bit 5 of ADM2 register)
  0: For two wire application
  1: For three wire application

- ACTDIS (bit 7 of PWCTL2)
  0: Active power calculation enable
  1: Active power calculation disable

- READIS (bit 6 of PWCTL1)
  0: Reactive power calculation enable
  1: Reactive power calculation disable

- APPDIS (bit 5 of PWCTL1)
  0: Apparent power calculation and ampere-hour ACC enable
  1: Apparent power calculation and ampere-hour ACC disable

- APPRMSCFCON (bit 2 of PWCTL1)
  0: Apparent power is accumulated
  1: Ampere-hour is accumulated

Table 19-11. List of Control Active Power and Energy

PWREN	CHMD	ACTDIS	Active power 1	Active power 2	Active energy
0	0/1	0/1	x	x	x
1	0/1	1	x	x	x
1	0	0	√	x	√
1	1	0	√	√	√

Table 19-12. List of Control Reactive Power and Energy

PWREN	CHMD	READIS	Reactive power 1	Reactive power 2	Reactive energy
0	0/1	0/1	x	x	x
1	0/1	1	x	x	x
1	0	0	√	x	√
1	1	0	√	√	√

Table 19-13. List of Control Apparent Power, Energy, RMS, and Ampere-hour ACC

PWREN	CHMD	APPDIS	APPRMS CFCON	Apparent power 1	Apparent power 2	Apparent energy	I1RMS	I2RMS	V1RMS	V2RMS	ampere-hour ACC
0	0/1	0/1	0/1	x	x	x	x	x	x	x	x
1	0	1	0/1	x	x	x	√	√	x	x	x
1	0	0	0	√	x	√	√	√	√	x	x
1	0	0	1	x	x	x	√	√	√	x	√
1	1	1	0/1	x	x	x	√	√	x	x	x
1	1	0	0	√	√	√	√	√	√	√	x
1	1	0	1	x	x	x	√	√	√	√	√

19.4.11 Notes on power calculation circuit

<R> (1) The performance of the power calculation circuit is only guaranteed when the main clock is running at 10 MHz and the data rate is 10 MHz/(128 × 18) = 4.34 kHz.

(2) Many registers in the power calculation circuit contain 2 bytes or more. When reading from a register that contains 2 bytes or more, read the lower bytes first and then the higher bytes. Similarly, when writing to a register that contains 2 bytes or more, write the lower bytes first and then the higher bytes. Other access methods might not work correctly.

(3) When using the power-saving mode, do not perform waveform sampling on the corresponding channel.

(4) After specifying the power-saving mode, the corresponding energy accumulator does not change. Therefore, after specifying this mode, do not read the energy register for the corresponding channel.

(5) Before specifying the power-saving mode, the source for the digital frequency conversion circuit must be checked or changed to ensure that the source of the digital frequency conversion circuit is not delivered by the corresponding channel. For example, before ACTDIS is set, “00” is prohibited for [CFSEL1, CFSEL0] (When “00” is specified, active power is selected for the source of the digital frequency conversion circuit).

(6) There is a recommended waiting time after setting power-on (ADPON bit of ADM2 register = 1) and 24-bit ΔΣ-type A/D conversion operation enable (ADCE2 bit of ADM2 register = 1). After the waiting time elapses, enable the operation of the power calculation circuit (PWREN bit of PWCTL1 register = 1)[Note].
When the operation of integrator is disabled (both of INTE1 bit and INTE2 bit of PWCTL1 register = 0), the recommended waiting time is 5 seconds.
When the operation of integrator is enabled (both of INTE1 bit and INTE2 bit of PWCTL1 register = 1), the recommended waiting time is 20 seconds.
The calculation accuracy is not guaranteed during the waiting time. Also disable the extended SFR (3rd SFR) interrupt during the waiting time. After the waiting time elapses, clear the interrupt flag and enable the interrupt.

Note After enabling the A/D conversion operation, the time, which is needed to implement the calculation that guarantees the accuracy, depends on the analog input state at the moment. It is because the digital filter stabilizing time varies depending on the analog input state. Evaluate the stabilizing time properly according to the usage environment.

CHAPTER 20 POWER QUALITY MEASUREMENT CIRCUIT

The power quality measurement circuit is used to measure the quality of input signals.

20.1 Power Quality Measurement Circuit Functions

The power quality measurement circuit has the following functions:

- Zero-crossing detection
- Zero-crossing timeout detection
- SAG detection
- Peak measurement
- Period and frequency measurement
- Fault detection
- Current channel gain adjustment

20.1.1 Generated interrupt signals

The power quality measurement circuit generates the following interrupt signals.

Table 20-1. Interrupt Signal of Power Measurement Quality Circuit

Interrupt Request Signal	Interrupt Request Flag	Interrupt Mask Flag	Interrupt Source
INTZX1	ZX1IF	ZX1MK	Zero-crossing detection interrupt for voltage channel 1
INTZX2	ZX2IF	ZX2MK	Zero-crossing detection interrupt for voltage channel 2
INTZXTO1	ZXTO1IF	ZXTO1MK	Zero-crossing timeout interrupt for voltage channel 1
INTZXTO2	ZXTO2IF	ZXTO2MK	Zero-crossing timeout interrupt for voltage channel 2
INTSAG1	SAG1IF	SAG1MK	SAG detection interrupt for voltage channel 1
INTSAG2	SAG2IF	SAG2MK	SAG detection interrupt for voltage channel 2
INTPKI1	PKI1IF	PKI1MK	Peak detection interrupt for current channel
INTPKV1	PKV1IF	PKV1MK	Peak detection interrupt for voltage channel
INTFAULTSIGN	FAULTSIGNIF	FAULTSIGNMK	Fault detection interrupt

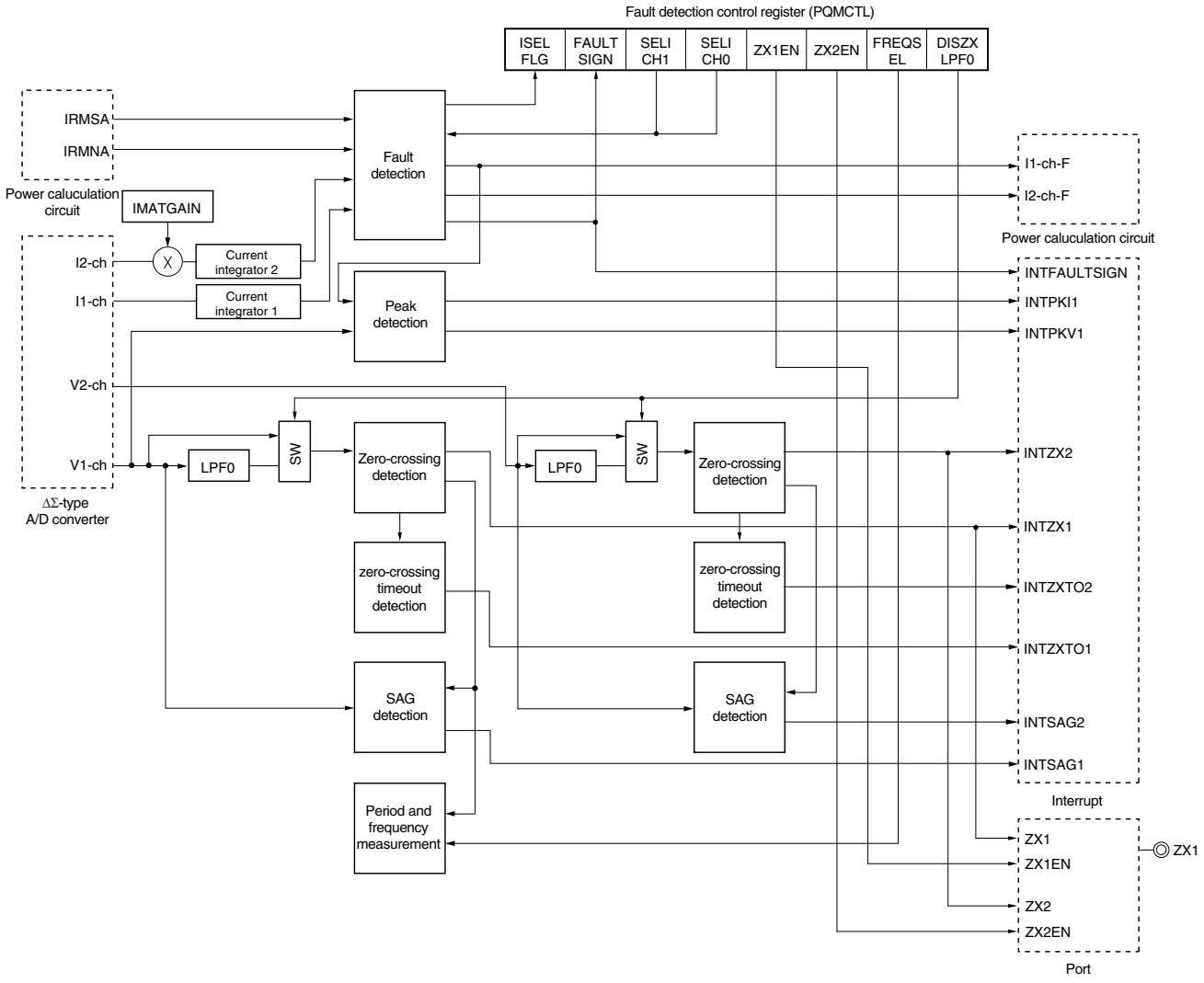
20.2 Configuration of Power Quality Measurement Circuit

The power quality measurement circuit includes the following hardware.

Table 20-2. Configuration of Power Quality Measurement Circuit

Item	Configuration
Controller	Zero-crossing detection circuit
	Zero-crossing timeout detection circuit
	SAG detection circuit
	Peak measurement circuit
	Period and frequency measurement circuit
	Fault detection circuit
	Current channel gain adjustment circuit
Registers	Period and frequency measurement result register (PFVAL)
	Peak current value register (IMAX)
	Peak current value clearing register (RSTIMAX)
	Peak voltage value register (VMAX)
	Peak voltage value clearing register (RSTVMAX)
Control registers	Zero-crossing timeout specification registers for voltage channels 1 and 2 (ZXTOUT1, ZXTOUT2)
	SAG line cycle number specification registers for voltage channels 1 and 2 (SAGNUM1, SAGNUM2)
	SAG level specification registers for voltage channels 1 and 2 (SAGVAL1, SAGVAL2)
	Peak current level specification register (IPKLMT)
	Peak voltage level specification register (VPKLMT)
	Gain specification register (IMATGAIN)
	Fault detection control register (PQMCTL)
	Fault detection threshold value specification register (IST)
	Fault control register (ICHK)

Figure 20-1. Block Diagram of Power Quality Measurement Circuit



(1) Zero-crossing detection circuit, zero-crossing timeout circuit

These circuits are used to monitor the status of the voltage signal input to the V1 and V2 channels.
Zero-crossing signal output is enabled by setting the ZX1EN and ZX2EN bits of the PQMCTL register to 1.

(a) Zero-crossing detection

For the V1 and V2 channel values input from the 24-bit ΔΣ-type A/D converter, the inversion of the sign (a zero crossing) is detected. It is possible to specify that a zero-crossing interrupt (INTZXn) occur when the sign is inverted.

(b) Zero-crossing timeout

If no zero-crossing is detected within the specified timeout period, a zero-crossing timeout interrupt (INTZXTOn) occurs.
The timeout time is specified for the ZXTOUTn register as a number of sampling clock (fs = 4.34 kHz) counts.

(2) SAG detection circuit

This circuit is used to monitor for changes in the input voltage (on the V1 and V2 channels).
If the SAG detection voltage level specified for the SAGVALn register or a lower level is continuously applied during the period of the number of line cycles specified for the SAGNUMn register, a SAG interrupt (INTSAGn) occurs.

(3) Peak measurement circuit

This circuit is used to monitor for peaks in the input voltage (on the V1 channel) and the input current (on the I1 channel).
If the absolute value of the input voltage (on the V1 channel) or the input current (on the I1 channel) exceeds the peak detection voltage or current level specified for the VPKLMT or IPKLMT register, an interrupt (INTPKV1 or INTPKI1) occurs, and the detected peak value (an absolute value) is stored in the VMAX or IMAX register.

The following two modes are available for reading peak values, and registers are prepared for each reading mode:

- Normal reading (register names: VMAX and IMAX)
  Detected peak values are read.
- Reading with a reset (register names: RSTVMAX and RSTIMAX)
  Detected peak values are read. After reading, the register values are cleared.

Remark n= 1, 2

(4) Period and frequency measurement circuit

This circuit is used to measure the period and frequency of the voltage channel (only the V1 channel). The period and frequency can be switched using the PQMCTL register.

(a) Period measurement

The sampling clock (fs = 4.34 kHz) of the 24-bit ΔΣ-type A/D converter is used to count a 20-line interval. This count result is stored in the PFVAL register.

(b) Frequency measurement

The sampling clock (fs = 4.34 kHz) of the 24-bit ΔΣ-type A/D converter is used to count a 20-line interval. This count value is used to calculate the frequency. The calculation result is stored in the PFVAL register as 0.0625 Hz/LSB.

(5) Fault detection circuit

This circuit is only used in the single-phase two-wire mode. I1 is connected to the line side and I2 is connected to the neutral side to perform current detection by using two lines.

(a) Fault detection

If the following conditions are met, a fault is detected, the active channel is switched, and an interrupt can be generated.

- The difference between the average current values (RMS values) for the I1 and I2 channels exceeds the multiple[Note] (1/8, 1/16, 1/32, or 1/64) specified for the average active channel current value.
- The larger of the average current values (RMS values) for the I1 and I2 channels exceeds the value specified for the IST register.

Note This multiple is specified for the ICHK register.

(b) Active channel switching

If the automatic mode is specified for the PQMCTL register and a fault condition is detected, the active channel is automatically switched. Switching the channel can also be specified.

The PQMCTL register can be used to check the current active channel.

(c) Fault interrupt occurrence

An interrupt (INTFAULTSIGN) occurs when a fault is detected or when the system returns from the fault status.

(6) Current channel gain correction

To detect faults, a comparison is made using the difference between current I1 and current I2. Therefore, current I1 and current I2 must be adjusted to the same amount of current in advance. The current gain correction circuit performs correction by adding gain to input current value I2.

20.3 Power Quality Measurement Circuit

The power quality measurement circuit uses the following registers.

These registers are all allocated to the extended SFR (3rd SFR) space.

For details about how to access the extended SFR (3rd SFR) space, see **CHAPTER 14 EXTENDED SFR (3rd SFR) INTERFACE**.

- Period and frequency measurement result register (PFVAL)
- Zero-crossing timeout specification registers for voltage channels 1 and 2 (ZXTOUT1, ZXTOUT2)
- SAG line cycle number specification registers for voltage channels 1 and 2 (SAGNUM1, SAGNUM2)
- SAG level specification registers for voltage channels 1 and 2 (SAGVAL1, SAGVAL2)
- Peak current level specification register (IPKLMT)
- Peak voltage level specification register (VPKLMT)
- Peak current value register (IMAX)
- Peak current value clearing register (RSTIMAX)
- Peak voltage value register (VMAX)
- Peak voltage value clearing register (RSTVMAX)
- Gain specification register (IMATGAIN)
- Fault detection control register (PQMCTL)
- Fault detection threshold value specification register (IST)
- Fault control register (ICHK)

(1) Period and frequency measurement result register (PFVAL)

Period and frequency measurement result of voltage channel 1 is stored in this register.

PFVAL is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 0000H.

Figure 20-2. Format of Period and Frequency Measurement Result Register (PFVAL)

Address: 100H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
PFVALL	PFVAL7	PFVAL6	PFVAL5	PFVAL4	PFVAL3	PFVAL2	PFVAL1	PFVAL0

Address: 101H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
PFVALH	PFVAL15	PFVAL14	PFVAL13	PFVAL12	PFVAL11	PFVAL10	PFVAL9	PFVAL8

PFVAL15 to 0	Used to store the Period and frequency measurement result of voltage channel 1
0000H to FFFFH	Period and frequency measurement result

Remark The FREQSEL bit of the PQMCTL register can be used to specify whether to store the result of measuring the period or frequency in the register.

- If FREQSEL = 0, the result of measuring the period is stored in the register.
- If FREQSEL = 1, the result of measuring the frequency is stored in the register.

(2) Zero-crossing timeout specification registers for voltage channels 1 and 2 (ZXTOUT1, ZXTOUT2)

The ZXTOUT1 and ZXTOUT2 registers are used to set the zero-crossing timeout specification of voltage channels 1 and 2.

ZXTOUT1 and ZXTOUT2 are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 03FFH.

Figure 20-3. Format of Zero-crossing Timeout Specification Registers for Voltage Channels 1 and 2 (ZXTOUT1, ZXTOUT2)

(a) ZXTOUT1

Address: 102H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ZXTOUT1L	ZXTOUT17	ZXTOUT16	ZXTOUT15	ZXTOUT14	ZXTOUT13	ZXTOUT12	ZXTOUT11	ZXTOUT10

Address: 103H After reset: 03H R/W

Symbol	7	6	5	4	3	2	1	0
ZXTOUT1H	0	0	0	0	0	0	ZXTOUT19	ZXTOUT18

(b) ZXTOUT2

Address: 104H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ZXTOUT2L	ZXTOUT27	ZXTOUT26	ZXTOUT25	ZXTOUT24	ZXTOUT23	ZXTOUT22	ZXTOUT21	ZXTOUT20

Address: 105H After reset: 03H R/W

Symbol	7	6	5	4	3	2	1	0
ZXTOUT2H	0	0	0	0	0	0	ZXTOUT29	ZXTOUT28

ZXTOUTn9 to n0	Specify the zero-crossing timeout specification of voltage channels 1 and 2 (n = 1, 2)
0000H to 03FFH	zero-crossing timeout specification

Cautions 1. Be sure to clear bits 2 to 7 of the ZXTOUT1H and ZXTOUT2H to 0.

2. When CHMD (bit 5 of ADM2) = 0, ZXTOUT2 cannot be written. The read value is the reset value.

(3) SAG line cycle number specification registers for voltage channels 1 and 2 (SAGNUM1, SAGNUM2)

SAGNUM1 and SAGNUM2 registers are used to set the line cycle number of the SAG detection for voltage channels 1 and 2.

These registers can be set by a half line unit.

SAGNUM1 and SAGNUM2 are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to FFH.

Figure 20-4. Format of SAG Line Cycle Number Specification Registers for Voltage Channels 1 and 2 (SAGNUM1, SAGNUM2)

(a) SAGNUM1

Address: 106H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
SAGNUM1	SAGNUM17	SAGNUM16	SAGNUM15	SAGNUM14	SAGNUM13	SAGNUM12	SAGNUM11	SAGNUM10

(b) SAGNUM2

Address: 10AH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
SAGNUM2	SAGNUM27	SAGNUM26	SAGNUM25	SAGNUM24	SAGNUM23	SAGNUM22	SAGNUM21	SAGNUM20

SAGNUMn7 to n0	Specify line cycle number of SAG detection for voltage channel n (n = 1, 2)
00H to FFH	Line cycle number of SAG detection

Caution When CHMD (bit 5 of ADM2) = 0, SAGNUM2 cannot be written. The read value is the reset value.

(4) SAG level specification registers for voltage channels 1 and 2 (SAGVAL1, SAGVAL2)

SAGVAL1 and SAGVAL2 registers are used to set the voltage level of the SAG detection for voltage channels 1 and 2.

SAGVAL1 and SAGVAL2 are allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up these registers.

Reset signal generation clears these registers to 000000H.

Figure 20-5. Format of SAG Level Specification Registers for Voltage Channels 1 and 2 (SAGVAL1, SAGVAL2)

(a) SAGVAL1

Address: 107H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SAGVAL1L	SAGVAL17	SAGVAL16	SAGVAL15	SAGVAL14	SAGVAL13	SAGVAL12	SAGVAL11	SAGVAL10

Address: 108H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SAGVAL1M	SAGVAL115	SAGVAL114	SAGVAL113	SAGVAL112	SAGVAL111	SAGVAL110	SAGVAL19	SAGVAL18

Address: 109H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SAGVAL1H	0	0	SAGVAL121	SAGVAL120	SAGVAL119	SAGVAL118	SAGVAL117	SAGVAL116

(b) SAGVAL2

Address: 10BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SAGVAL2L	SAGVAL27	SAGVAL26	SAGVAL25	SAGVAL24	SAGVAL23	SAGVAL22	SAGVAL21	SAGVAL20

Address: 10CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SAGVAL2M	SAGVAL215	SAGVAL214	SAGVAL213	SAGVAL212	SAGVAL211	SAGVAL210	SAGVAL29	SAGVAL28

Address: 10DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SAGVAL2H	0	0	SAGVAL221	SAGVAL220	SAGVAL219	SAGVAL218	SAGVAL217	SAGVAL216

SAGVALn21 to n0	Specify voltage level of SAG detection for voltage channel n (n = 1, 2)
000000H to 3FFFFFH	Voltage level of SAG detection

Cautions 1. Be sure to clear bits 6 and 7 of the SAGVAL1H and SAGVAL2H to 0.

2. When CHMD (bit 5 of ADM2) = 0, SAGVAL2 cannot be written. The read value is the reset value.

(5) Peak current level specification register (IPKLMT)

IPKLMT register is used to set the current level of the peak detection for current channel 1.

IPKLMT is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to FFFFH.

Figure 20-6. Format of Peak Current Level Specification Register (IPKLMT)

Address: 10EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IPKLMTL	IPKLMT7	IPKLMT6	IPKLMT5	IPKLMT4	IPKLMT3	IPKLMT2	IPKLMT1	IPKLMT0

Address: 10FH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
<R> IPKLMTM	IPKLMT15	IPKLMT14	IPKLMT13	IPKLMT12	IPKLMT11	IPKLMT10	IPKLMT9	IPKLMT8

IPKLMT15 to 0	Specify peak detection current level for current channel 1
0000H to FFFFH	Current level of peak detection

(6) Peak voltage level specification register (VPKLMT)

VPKLMT register is used to set the voltage level of the peak detection for voltage channel 1.

VPKLMT is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to FFFFH.

Figure 20-7. Format of Peak Voltage Level Specification Register (VPKLMT)

Address: 110H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
VPKLMTL	VPKLMT7	VPKLMT6	VPKLMT5	VPKLMT4	VPKLMT3	VPKLMT2	VPKLMT1	VPKLMT0

Address: 111H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
<R> VPKLMTM	VPKLMT15	VPKLMT14	VPKLMT13	VPKLMT12	VPKLMT11	VPKLMT10	VPKLMT9	VPKLMT8

VPKLMT15 to 0	Specify peak detection voltage level for voltage channel 1
0000H to FFFFH	Voltage level of peak detection

(7) Peak current value register (IMAX)

When the absolute value of current channel 1 exceeds the current peak detection level, the detected peak value (an absolute value) is stored in this register. This register value is not cleared after being read.

IMAX is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 000000H.

Figure 20-8. Format of Peak Current Value Register (IMAX)

Address: 112H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
IMAXL	IMAX7	IMAX6	IMAX5	IMAX4	IMAX3	IMAX2	IMAX1	IMAX0

Address: 113H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
IMAXM	IMAX15	IMAX14	IMAX13	IMAX12	IMAX11	IMAX10	IMAX9	IMAX8

Address: 114H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
IMAXH	0	0	IMAX21	IMAX20	IMAX19	IMAX18	IMAX17	IMAX16

IMAX21 to 0	Store the detected peak current value (an absolute value)
000000H to 3FFFFFH	Peak current value (not cleared after reading)

(8) Peak current value clearing register (RSTIMAX)

When the absolute value of current channel 1 exceeds the current peak detection level, the detected peak value (an absolute value) is stored in this register. This register value is cleared after being read.

RSTIMAX is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 000000H.

Figure 20-9. Format of Peak Current Value Clearing Register (RSTIMAX)

Address: 116H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RSTIMAXL	RSTIMAX7	RSTIMAX6	RSTIMAX5	RSTIMAX4	RSTIMAX3	RSTIMAX2	RSTIMAX1	RSTIMAX0

Address: 117H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RSTIMAXM	RSTIMAX15	RSTIMAX14	RSTIMAX13	RSTIMAX12	RSTIMAX11	RSTIMAX10	RSTIMAX9	RSTIMAX8

Address: 118H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RSTIMAXH	0	0	RSTIMAX21	RSTIMAX20	RSTIMAX19	RSTIMAX18	RSTIMAX17	RSTIMAX16

RSTIMAX21 to 0	Store the detected peak current value (an absolute value)
000000H to 3FFFFFH	Peak current value (cleared after reading)

(9) Peak voltage value register (VMAX)

When the absolute value of voltage channel 1 exceeds the current peak detection level, the detected peak value (an absolute value) is stored in this register. This register value is not cleared after being read.

VMAX is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 000000H.

Figure 20-10. Format of Peak Voltage Value Register (VMAX)

Address: 119H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
VMAXL	VMAX7	VMAX6	VMAX5	VMAX4	VMAX3	VMAX2	VMAX1	VMAX0

Address: 11AH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
VMAXM	VMAX15	VMAX14	VMAX13	VMAX12	VMAX11	VMAX10	VMAX9	VMAX8

Address: 11BH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
VMAXH	0	0	VMAX21	VMAX20	VMAX19	VMAX18	VMAX17	VMAX16

VMAX21 to 0	Store the detected peak voltage value (an absolute value)
000000H to 3FFFFFH	Peak voltage value (not cleared after reading)

(10) Peak voltage value clearing register (RSTVMAX)

When the absolute value of voltage channel 1 exceeds the current peak detection level, the detected peak value (an absolute value) is stored in this register. This register value is cleared after being read.

RSTVMAX is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 000000H.

Figure 20-11. Format of Peak Voltage Value Clearing Register (RSTVMAX)

Address: 11DH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RSTVMAXL	RSTVMAX7	RSTVMAX6	RSTVMAX5	RSTVMAX4	RSTVMAX3	RSTVMAX2	RSTVMAX1	RSTVMAX0

Address: 11EH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RSTVMAXM	RSTVMAX15	RSTVMAX14	RSTVMAX13	RSTVMAX12	RSTVMAX11	RSTVMAX10	RSTVMAX9	RSTVMAX8

Address: 11FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
RSTVMAXH	0	0	RSTVMAX21	RSTVMAX20	RSTVMAX19	RSTVMAX18	RSTVMAX17	RSTVMAX16

RSTVMAX21 to 0	Store the detected peak voltage value (an absolute value)
000000H to 3FFFFFH	Peak voltage value (cleared after reading)

(11) Gain specification register (IMATGAIN)

The IMATGAIN register is used to set the gain value of current channel 2.

IMATGAIN is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 0000H.

Figure 20-12. Format of Gain Specification Register (IMATGAIN)

Address: 120H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IMATGAINL	IMATGAIN7	IMATGAIN6	IMATGAIN5	IMATGAIN4	IMATGAIN3	IMATGAIN2	IMATGAIN1	IMATGAIN0

Address: 121H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IMATGAINH	0	0	0	0	IMATGAIN11	IMATGAIN10	IMATGAIN9	IMATGAIN8

IMATGAIN11 to 0	Specify the gain value of current channel 2
0000H to 0FFFH	Gain value

Cautions 1. Be sure to clear bit 4 to 7 of the IMATGAINH to 0.

2. When CHMD (bit 5 of ADM2) = 1, IMATGAIN cannot be written. The read value is the reset value.

(12) Fault detection control register (PQMCTL)

This register controls current channel selection, fault interrupt settings, switching between period and frequency measurement, and zero-crossing detection.

PQMCTL is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 20-13. Format of Fault Detection Control Register (PQMCTL)

Address: 122H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PQMCTL	ISELFLG	FAULTSIGN	SELICH1	SELICH0	ZX1EN	ZX2EN	FREQSEL	DISZXLPF0

ISELFLG	Active current channel during power measurement
0	Current channel 1
1	Current channel 2

FAULTSIGN	Specify the trigger for the fault detection interrupt (INTFAULTSIGN)
0	Triggers an interrupt when switching from the normal mode to the fault mode.
1	Triggers an interrupt when switching from the fault mode to the normal mode.

SELICH1	SELICH0	Specify the current channel used for power measurement
0	0	Automatically selects the current channel according to the fault status.
0	1	Selects current channel 1[Note] .
1	0	Selects current channel 2[Note] .
1	1	Automatically selects the current channel according to the fault status.

Note This setting is prohibited in the three-wire mode (CHMD = 1).

ZX1EN	Control zero-crossing output 1
0	Disables output of the zero-crossing 1 signal.
1	Enables output of the zero-crossing 1 signal.

ZX2EN	Control zero-crossing output 2
0	Disables output of the zero-crossing 2 signal.
1	Enables output of the zero-crossing 2 signal.

FREQSEL	Select period or frequency measurement
0	Stores the result of measuring the period in the PFVAL register.
1	Stores the result of measuring the frequency in the PFVAL register.

DISZXLPF0	Set up the low-pass filter used during zero-crossing detection
0	Enables the low-pass filter and outputs the zero-crossing signal ZX from LPF0.
1	Disables the low-pass filter and outputs the zero-crossing signal ZX from the 24-bit ΔΣ-type A/D converter.

(13) Fault detection threshold value specification register (IST)

The IST register is used to set the threshold value of fault detection.

IST is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 8BH.

Figure 20-14. Format of Fault Detection Threshold Value Specification Register (IST)

Address: 123H After reset: 8BH R/W

Symbol	7	6	5	4	3	2	1	0
IST	IST7	IST6	IST5	IST4	IST3	IST2	IST1	IST0

IST7 to 0 (00H to FFH)	Specify the fault detection threshold value (% fs)
01H	0.00 %
8BH	0.30 % (default)
FFH	0.55 %

The IST adjustment range is 0.00216% fs/LSB.

For example, if 0.3% of the full-scale value is specified for the threshold value, IST is calculated as follows:

- IST = 0.3/0.00216 = 8BH

If the ICHKEN bit of the ICHK register is set to 1 and the current signal is larger than the IST value, a fault can be detected. If the current signal does not reach the threshold value specified for IST, fault detection is automatically disabled.

(14) Fault control register (ICHK)

This register is used to control fault detection and specify detection conditions.

ICHK is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 84H.

Figure 20-15. Format of Fault Control Register (ICHK)

Address: 124H After reset: 84H R/W

Symbol	7	6	5	4	3	2	1	0
ICHK	ICHKEN	0	0	0	ICHK3	ICHK2	ICHK1	ICHK0

ICHKEN[Note]	Control current fault detection
0	Disables current fault check
1	Enables current fault check

ICHK3	ICHK2	ICHK1	ICHK0	Current difference threshold value
1	x	x	x	Current difference threshold value = 1/8
0	1	x	x	Current difference threshold value = 1/16 (default)
0	0	1	x	Current difference threshold value = 1/32
0	0	0	1	Current difference threshold value = 1/64
0	0	0	0	Current difference threshold value = 1/16

Caution Be sure to clear bits 4 to 6 to 0.

Note The fault detection operation depends on the settings for the ICHK and IST registers as follows.

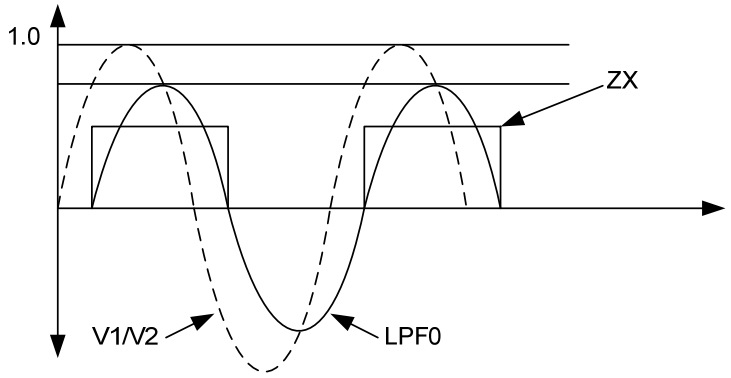
ICHIKEN	IST	ICHK3 to 0	Larger of Two Current Channel Values	Difference in Current Between Two Current Channels	Fault Detection Operation
0	x	xxxx	x	x	Fault detection unavailable
1	IST	xxxx	≤ IST	x	Fault detection unavailable
		1xxx	> IST	≤ 1/8	Normal status
				> 1/8	Fault status detected
		01xx		≤ 1/16	Normal status
				> 1/16	Fault status detected
		001x		≤ 1/32	Normal status
				> 1/32	Fault status detected
		0001		≤ 1/64	Normal status
				> 1/64	Fault status detected
		0000		≤ 1/16	Normal status
				> 1/16	Fault status detected

20.4 Power Quality Measurement Circuit Function Details

(1) Zero-crossing detection

Zero crossings are detected on the voltage channel. The input voltage is filtered using a low-pass filter (LPF0), and then zero-crossings are detected.

Figure 20-16. Zero-Crossing Detection Timing



The zero-crossing signal ZXn is output from LPF0 regardless of whether it is bypassed. LPF0 is a single-pole 68 Hz filter. Therefore, there is a phase delay of approximately 1.9 ms (at 50 Hz) between the LPF0 input and output. Similarly, there is a phase delay of approximately 1.1 ms (at 50 Hz) between the analog input and A/D converter output for the software block (the digital filter) of the 24-bit ΔΣ-type A/D converter. Due to the phase delay of LPF0 and the A/D converter, there is a delay of approximately 1.1 ms (when bypassing LPF0) or 3.02 ms (when using LPF0) between the analog input and output of the zero-crossing signal ZXn on the voltage channel.

Table 20-3. Zero-Crossing Detection Delay Time

	Condition (at 50 Hz)	
	When Bypassing LPF0	When Using LPF0
Delay between the analog input and output of the zero-crossing signal ZXn on the voltage channel	1.1 ms	3.02 ms

Remark n = 1, 2

(2) Zero-crossing timeout

The zero-crossing timeout function generates an interrupt if no zero-crossing is detected for a fixed period of time. The zero-crossing timeout function can be used separately for the V1 and V2 channels.

When a zero crossing is detected by the zero-crossing detection circuit, the value of the ZXOUTn register is read to an internal counter. The counter value is decremented for each sampling clock (fs). If the counter value reaches 0 before the next zero crossing is detected, a zero-crossing timeout interrupt (INTZXTOn) occurs. Note that the default value of the ZXOUTn register is 3FFH, and up to 0.23 s can be counted.

Figure 20-17. Timing of Zero-crossing Timeout Interrupt (INTZXTOn)

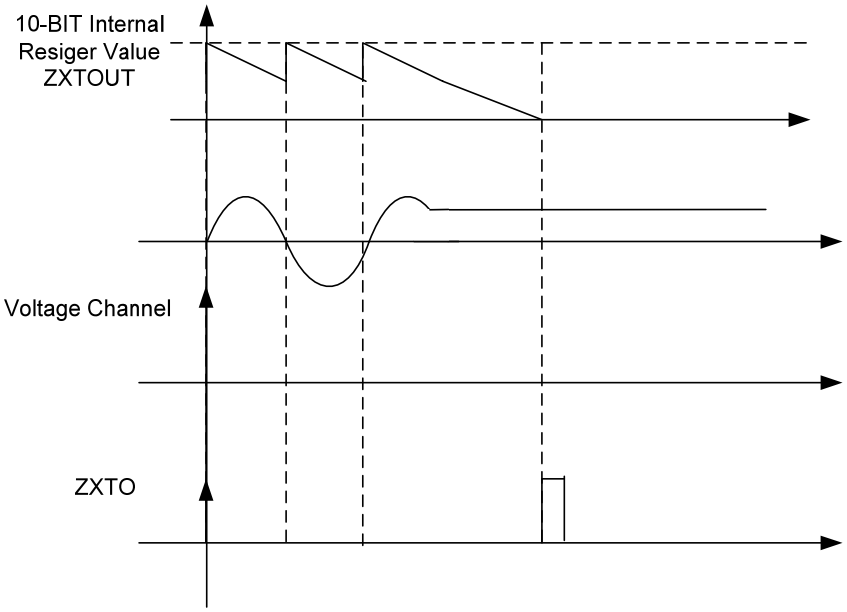
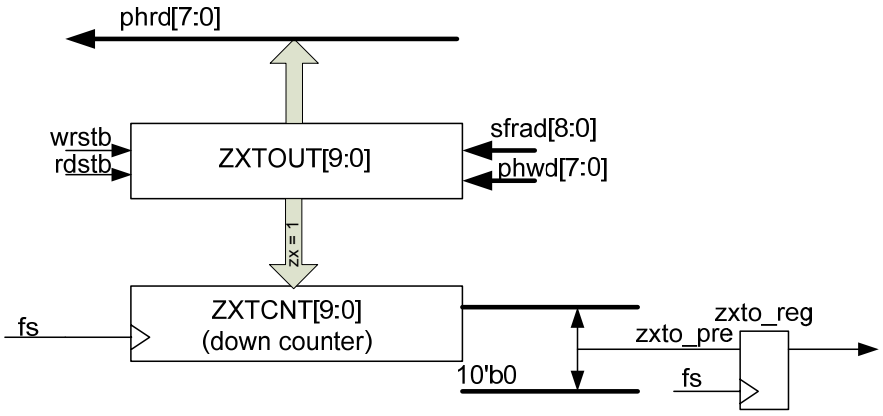


Figure 20-18. Internal Counter of ZXOUTn Register



Remark n = 1, 2

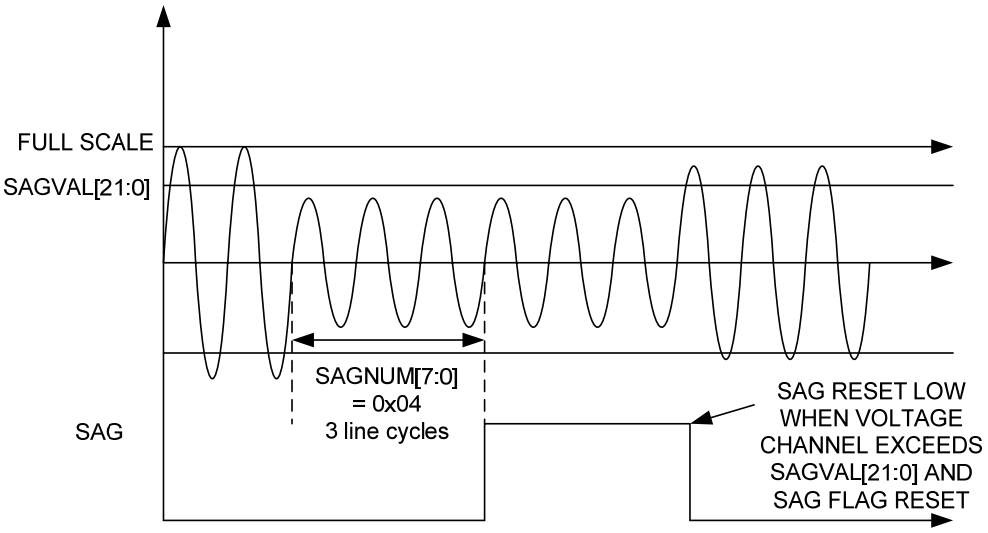
(3) SAG detection function

This function detects input voltage SAG. SAG can be detected separately for the V1 and V2 input voltages.

If the absolute value of the input voltage (23 bits) is less than or equal to the voltage value specified by the SAGVALn register for the duration specified by the SAGNUMn register, a SAG interrupt signal (INTSAGn) is generated.

The SAGNUMn register has an 8-bit format, and up to 255 line cycles can be specified for it. Note that the SAGVALn register has a 22-bit format.

Figure 20-19. Timing of SAG Detection



Remark n = 1, 2

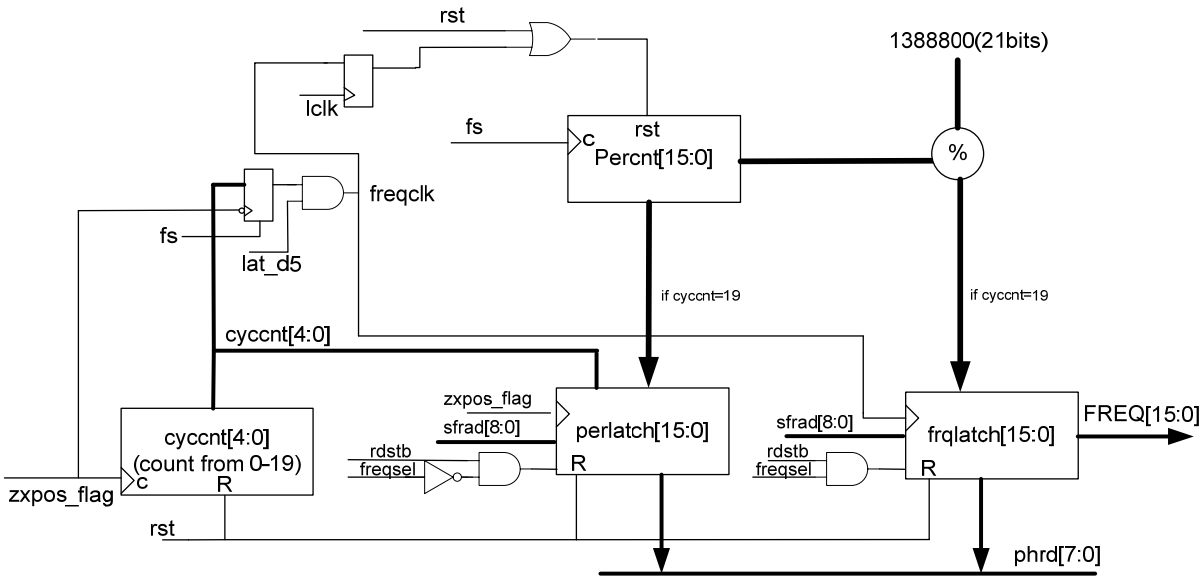
(4) Period and frequency measurement

The line cycle and frequency of the voltage channel (V1) can be measured. The PQMCTL register is used to specify whether to measure the line cycle or frequency. This measurement is performed for 20 line cycles at a time.

During cycle measurement, the fs (4.34 kHz) clock is used to count for 20 line cycles. Therefore, 11.5 μs/LSB (0.07% precision) is used. If the line frequency is 60 Hz, the counter value is 1,447.

During frequency measurement, the counter value is used to perform a calculation for 20 line cycles. If the line frequency is 60 Hz, the precision is 0.0625 Hz/LSB, and the calculation result is 960.

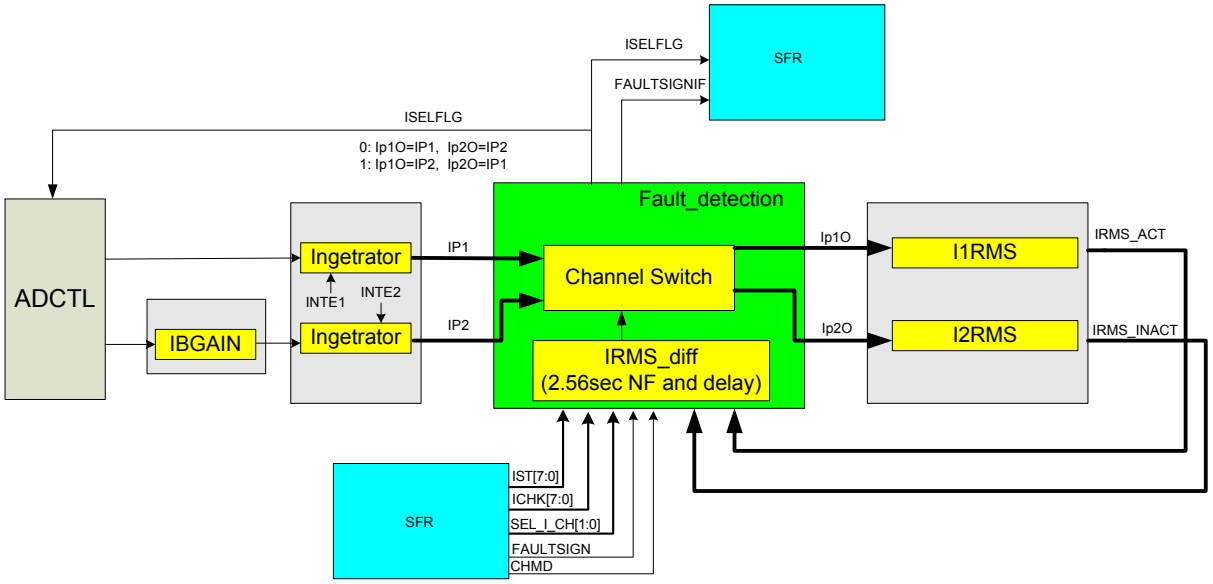
Figure 20-20. Period and Frequency Measurement Circuit



(5) Fault detection

Fault detection can be performed only in the single-phase two-wire mode. By connecting I1 to the line side and I2 to the neutral side, two lines are used to detect the current. The fault detection circuit is positioned before the power calculation circuit.

Figure 20-21. Fault Detector

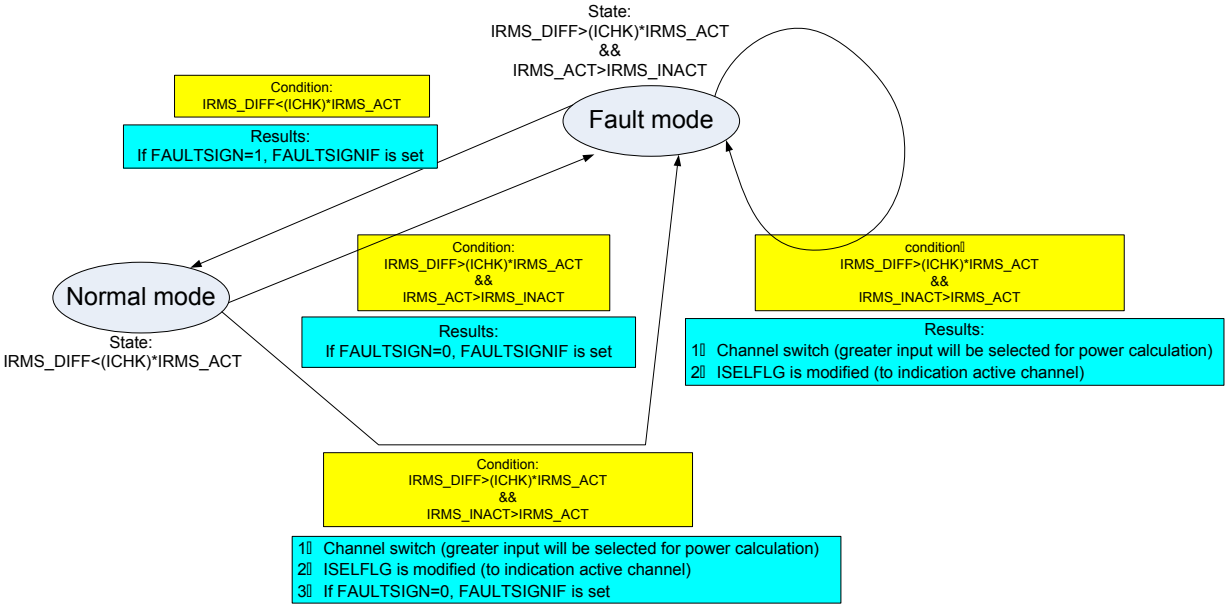


If the difference in current between Ip1 and Ip2 is greater than the multiple (1/8, 1/16, 1/32, or 1/64) specified for the active channel, a fault is detected. If a fault condition is detected and the current value for the inactive channel is larger than that for the active channel, the channels are switched, and the channel that has the larger current value automatically becomes the active channel. The active channel can be checked using the ISELFLG bit of the PQMCTL register. If this bit is 0, Ip1 channel is selected as the active channel, and, if the bit is 1, Ip2 channel is selected. This bit is automatically updated when the channel is switched.

The active channel can also be specified by using the SELICH1 and SELICH0 bits (a two-bit format) of the PQMCTL register. If both bits are set to either 0 or 1, the active channel is switched according to fault detection.

The following figure shows the transitions between the normal mode and fault mode.

Figure 20-22. Transitions Between the Normal Mode and Fault Mode



Remark IRMS_ACT: RMS value of the current channel used for power measurement
IRMS_INACT: RMS value of the current channel not used for power measurement
IRMS_DIFF: ABS (IRMS_ACT-IRMS_INACT)
IST: Fault detection threshold value
ICHK: Current difference threshold value
FAULTSIGN: Bit for specifying the fault interrupt trigger
FAULTSIGNIF: Fault mode change interrupt flag

Note that fault detection is automatically disabled for values less than the value specified by the IST register. This is to avoid misdetection for small amounts of current.

Because the difference between the Ip1 and Ip2 current signals is detected, the current value for the two channels must be corrected in advance. To perform this correction, the correction value must be specified for the IMATGAIN register. For details, see **(7) Current channel gain correction**.

To avoid incorrect power measurement when a fault event occurs, the input current is always monitored.
After a reset, the active channel is set to Ip1 by default, and this channel is used to calculate the active power, reactive power, and apparent power.

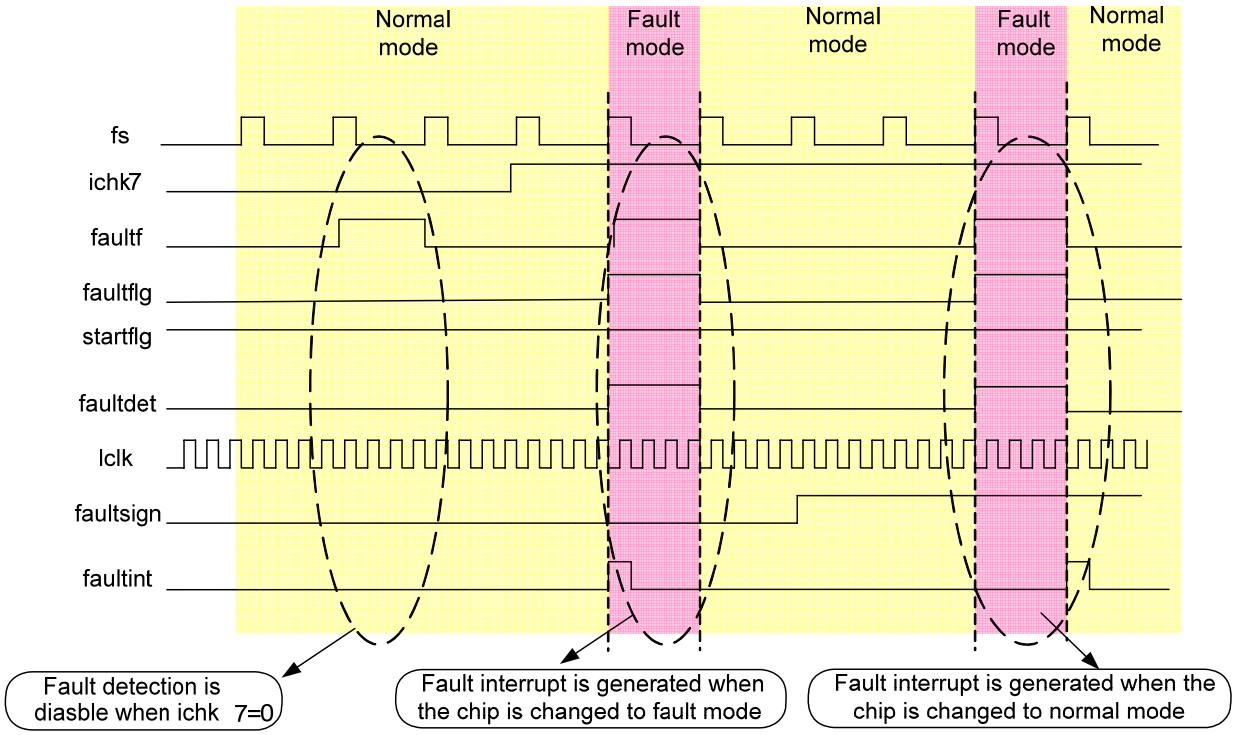
Remarks 1. To guarantee fault detection accuracy, it is recommended to set ZXRMS (bit 6 of the NLCTL register) to 1 if the fault detection current difference restriction is set to 1/64 or 1/32 (by using bits 3 to 0 of the ICHK register).

2. Fault detection is disabled in the single-phase three-wire mode (CHMD = 1).

If the FAULTSIGN bit of the PQMCTL register is cleared to 0, an interrupt occurs when a fault condition is detected (when the system transitions from the normal status to the fault status). If 1 is specified, an interrupt occurs when the system returns from the fault status (when the system transitions from the fault status to the normal status).

The fault interrupt and channel switching signals are passed through a filter to avoid misdetection. Therefore, there is a delay of approximately three seconds when a fault event occurs.

Figure 20-23. Fault Interrupt Timing



(6) Peak detection

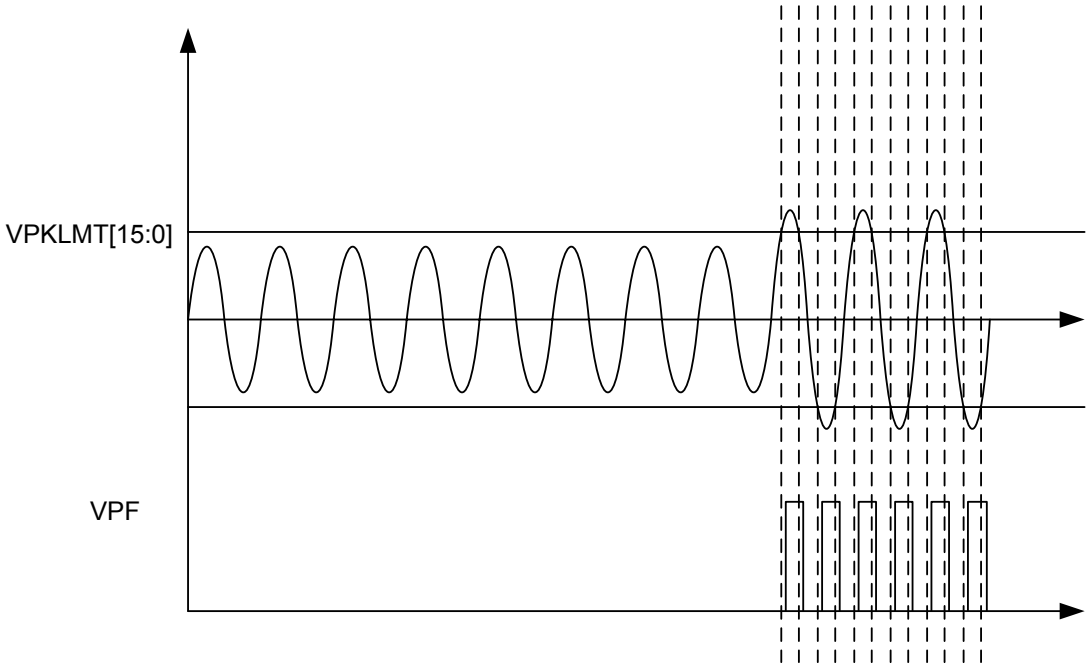
If the absolute value of the higher 16 bits of the voltage channel exceeds the peak voltage value (the VPKLMT register value), an INTPKV1 interrupt occurs.

If the absolute value of the higher 16 bits of the current channel exceeds the peak current value (the IPKLMT register value), an INTPKI1 interrupt occurs.

The maximum absolute value of the voltage channel is saved in the internal VMAX register, and the value can be read from the VMAX register in the extended SFR (3rd SFR) space. If the RSTVMAX register is used to read this value, the register value is cleared to 0 after being read.

The maximum absolute value of the current channel is saved in the internal IMAX register, and the value can be read from the IMAX register in the extended SFR (3rd SFR) space. If the RSTIMAX register is used to read this value, the register value is cleared to 0 after being read.

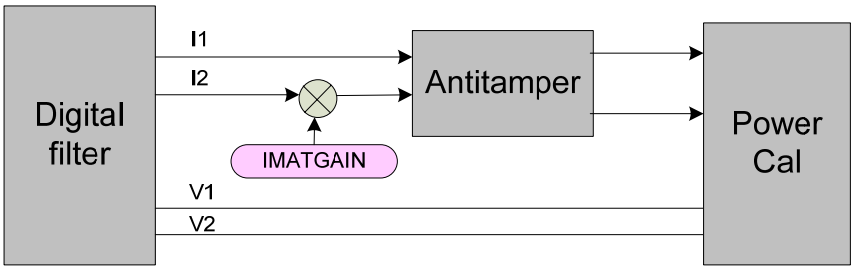
Figure 20-24. Peak Detection Timing



(7) Current channel gain correction

To detect faults, a comparison is made using the difference between current I1 and current I2. Therefore, current I1 and current I2 must be adjusted to the same amount of current in advance. The current gain correction circuit performs correction by adding gain to input current value I2.

Figure 20-25. Current Gain Correction Circuit



$$output\_IMATGAIN = input\_IMATGAIN \times (1 + \frac{IMATGAIN}{2^{12}})$$

IMATGAIN is a 12-bit signed integer register. The gain resolution for this register is 0.0244% LSB (2^{-12} = 0.0244%).

20.5 Notes on Power Quality Measurement Circuit

The registers for the power quality measurement circuit are allocated to the extended SFR (3rd SFR) space, and some of them contain 2 bytes or more. When reading from a register that contains 2 bytes or more, be sure to read the entire register and read the lower bytes first. Similarly, when writing to a register that contains 2 bytes or more, be sure to write to the entire register and write the lower bytes first. Reading and writing are not possible using other methods.

The target 2-byte registers are as follows.

Table 20-4. 2-byte Register List

Name	R/W	Read Buffer	Write Buffer
Period and frequency measurement result register (PFVAL)	R	Y	–
Zero-crossing timeout specification registers for voltage channel 1 (ZXTOUT1)	R/W	–	Y
Zero-crossing timeout specification registers for voltage channel 2 (ZXTOUT2)	R/W	–	Y
SAG level specification registers for voltage channel 1 (SAGVAL1)	R/W	–	Y
SAG level specification registers for voltage channel 2 (SAGVAL2)	R/W	–	Y
Peak current level specification register (IPKLMT)	R/W	–	Y
Peak voltage level specification register (VPKLMT)	R/W	–	Y
Peak current value register (IMAX)	R	Y	–
Peak current value clearing register (RSTIMAX)	R	Y	–
Peak voltage value register (VMAX)	R	Y	–
Peak voltage value clearing register (RSTVMAX)	R	Y	–
Gain specification register (IMATGAIN)	R/W	–	Y

CHAPTER 21 DIGITAL FREQUENCY CONVERSION CIRCUIT

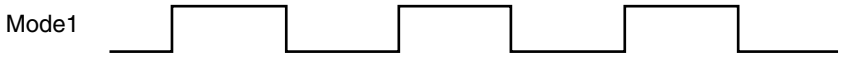
21.1 Digital Frequency Conversion Circuit Functions

The digital frequency conversion circuit integrates the measured active power, reactive power, and apparent power/average current (an RMS value), and then outputs the result as a pulse. There are two types of output pulse waveforms, and the frequency can be programmed by manipulating the CFMUL register.

(1) Mode 1

Output in the range from 0 to 135 Hz is possible.
A waveform that has a pulse duty ratio of 50:50 is output.
However, if the cycle exceeds 180 ms, the high-level width is fixed to 90 ms.

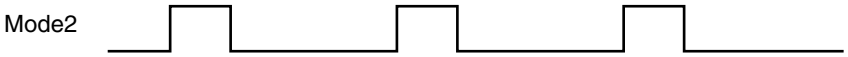
Figure 21-1. Output Pulse Waveform (Mode 1)



(2) Mode 2

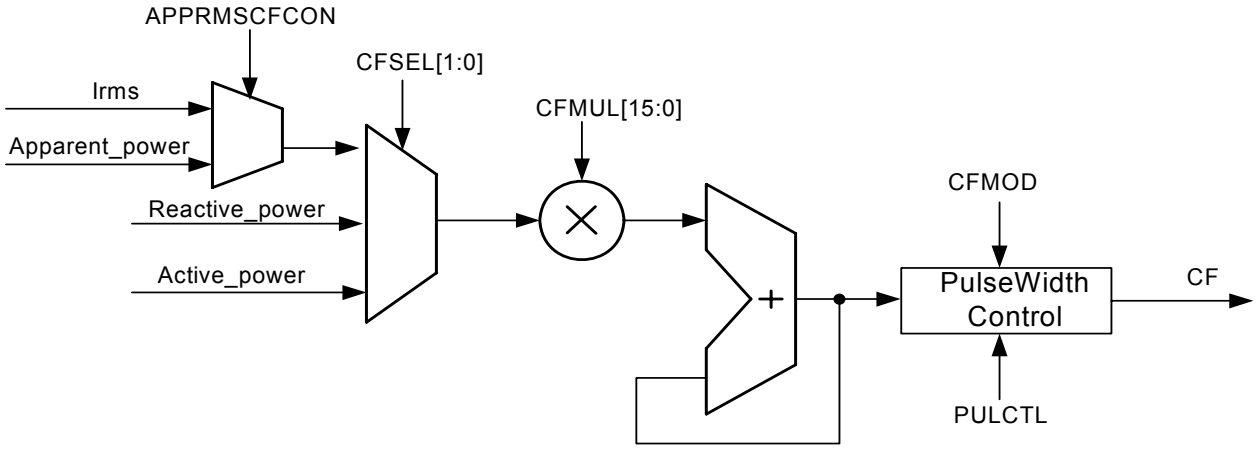
Output in the range from 0 to 1,085 Hz is possible.
A pulse waveform that has a fixed high-level width is output. The high-level width can be specified for the PULCTL register as 30, 60, 90, or 180 μs.
This mode is mainly used in Japan.

Figure 21-2. Output Pulse Waveform (Mode 2)



21.2 Configuration of Digital Frequency Conversion Circuit

Figure 21-3. Block Diagram of Digital Frequency Conversion Circuit



21.3 Registers Controlling Digital Frequency Conversion Circuit

The power quality measurement circuit is controlled by the following three types of registers.

These registers are all allocated to the extended SFR (3rd SFR) space.

For details about how to access the extended SFR (3rd SFR) space, see **CHAPTER 14 EXTENDED SFR (3RD SFR) INTERFACE**.

- Frequency conversion control register (CFCTL)
- Frequency scaling specification register (CFMULL, CFMULH)
- Pulse width specification register (PULCTL)

(1) Frequency conversion control register (CFCTL)

This register is used to control CF pulse output and select the output mode.

CFCTL is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 02H.

Figure 21-4. Format of Frequency Conversion Control Register (CFCTL)

Address: 150H After reset: 02H R/W

Symbol	7	6	5	4	3	2	1	0
CFCTL	0	0	0	0	CFSEL1	CFSEL0	DISCF	CFMOD

CFSEL1	CFSEL0	Select the CF pulse output (by selecting active power, reactive power, or apparent power/average current)
0	0	Outputs the active power as a CF pulse.
0	1	Outputs the reactive power as a CF pulse.
1	x	Outputs the apparent power/average current as a CF pulse.

DISCF	Enable or disable CF output
0	Enables CF output (A pulse is output from the CF pin).
1	Disables CF output (No pulse is output from the CF pin).

CFMOD	Select the CF output mode
0	Outputs the mode 1 waveform (which has a duty ratio of 50%)[Note].
1	Outputs the mode 2 waveform (which has a fixed high-level width).

Note If the output cycle exceeds 180 ms, the high-level width is fixed to 90 ms.

Caution Before specifying the power-saving mode, the source for the digital frequency conversion circuit must be checked or changed to ensure that the source of the digital frequency conversion circuit is not delivered by the corresponding channel. For example, before ACTDIS is set, "00" is prohibited for [CFSEL1, CFSEL0] (When "00" is specified, active power is selected for the source of the digital frequency conversion circuit).

(2) Frequency scaling specification register (CFMUL)

This register is used to specify the frequency scaling value.

CFMUL is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to FFH.

Figure 21-5. Format of Frequency Scaling Specification Register (CFMUL)

Address: 151H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
CFMULL	CFMUL7	CFMUL6	CFMUL5	CFMUL4	CFMUL3	CFMUL2	CFMUL1	CFMUL0

Address: 152H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
CFMULH	CFMUL15	CFMUL14	CFMUL13	CFMUL12	CFMUL11	CFMUL10	CFMUL9	CFMUL8

CFMUL15 to 0	Specify the CF output pulse width
0000H to FFFFH	Specifies the scaling value for the CF output pulse.

Cautions 1. Only change the value specified for the CFMUL register while DISCF = 1.

2. Specifying 0000H for the CFMUL register is prohibited. If 0000H is specified, it is handled as FFFFH.

(3) Pulse width specification register (PULCTL)

This register is used to specify the high-level width of the pulse used during mode 2 operation.

PULCTL is allocated to the extended SFR (3rd SFR) space.

Use the extended SFR (3rd SFR) interface to set up this register.

Reset signal generation clears this register to 00H.

Figure 21-6. Format of Pulse Width Specification Register (PULCTL)

Address: 153H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PULCTL	0	0	0	0	0	0	PULCTL1	PULCTL0

PULCTL1	PULCTL0	Specify the high-level width of the pulse (at mode 2 operation)
0	0	30 μs
0	1	90 μs
1	0	60 μs
1	1	180 μs

Caution Only change the value specified for the PULCTL register while DISCF = 1.

21.4 Digital Frequency Conversion Circuit Functions

The digital frequency conversion circuit converts the measured power value to a pulse waveform. The circuit adds and accumulates power values calculated by the power calculation unit and outputs a pulse when the threshold value is exceeded.

<R> **Figure 21-7. Pulse Output (Mode 1)**

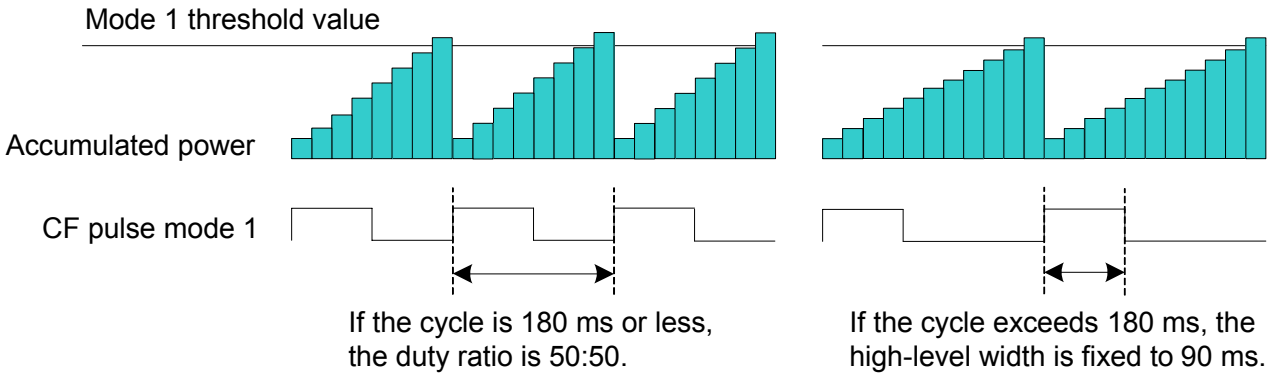
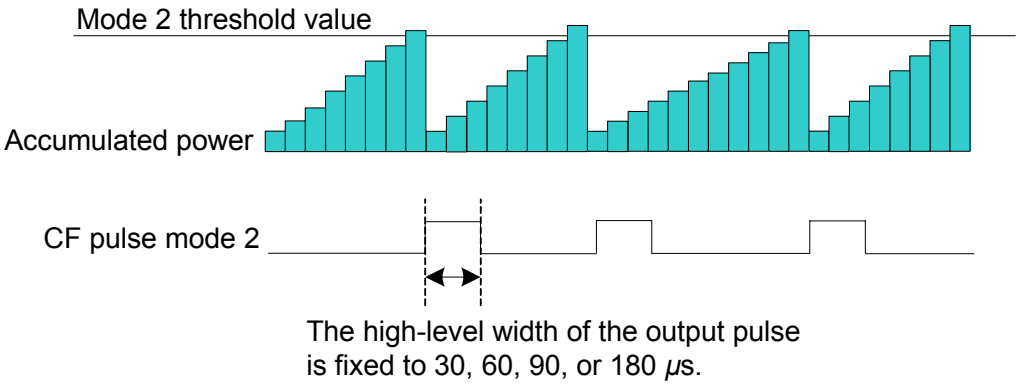


Figure 21-8. Pulse Output (Mode 2)



Cautions 1. In mode 1, if the cycle exceeds 180 ms, the high-level width is fixed to 90 ms.

2. For the same input voltage, the CF pulse frequency of mode 2 is eight times that of mode 1.

The power value for each sample is scaled and input to an accumulator. If the value in the accumulator exceeds the fixed threshold value, an overflow flag is set, and one pulse is output to the CF pin each time an overflow occurs. The threshold value for the accumulator is fixed, and the timing at which overflows occur can be adjusted by scaling the input power value to the value specified by the CFMUL register. A scaling value in the range from 1/(2^16) to (2^16-1)/(2^16) can be specified, in 1/(2^16) steps.

If the CFMUL register is set to 1, the input power value is scaled to 1/(2^16). If the CFMUL register is set to FFFFH, the input power value is scaled to (2^16-1)/(2^16).

(1) Input data selection method

The active power, reactive power, or apparent power/average current of the power calculation circuit can be selected as input to the digital frequency conversion circuit.

The CFSEL0 and CFSEL1 bits of the CFCTL register can be used to select which measurement result to output as a pulse. Only change these bits while the digital frequency conversion circuit is stopped (DISCF = 1).

(2) Value specified for the CFMUL register

The maximum value of CF output for full-scale input is 135.623 Hz (in mode 1) or 1084.984 Hz (in mode 2).

To obtain the rated CF frequency N Hz, the values specified for the CF frequency and CFMUL register have the following relationship.

(a) mode 1

CFMUL = 2^{16} x N / (135.623 x PowerRate)

(b) mode 2

CFMUL = 2^{16} x N / (1084.984 x PowerRate)

N : CF output frequency for rated input

PowerRate : Ratio of the rated power to the full-scale power

$$PowerRate = \frac{PowerGivenLoad}{PowerFullScale}$$

Remark When calculating PowerRate, use the same offset, gain, and scaling settings for the full-scale and rated input.

These settings can be specified using the following registers.

- Offset specification register : ACTnOS, REAnOS, InRMSOS, and VnRMSOS (n = 1, 2)
- Gain specification register : ACTnGAIN, REAnGAIN, APPnGAIN, and IRMSnGAIN (n = 1, 2)
- Scaling specification register : ACTDIV, READIV, and APPDIV

(3) CF output pulse

Two types of pulses can be output from CF pin. The pulse selection is set by using CFCTL register.

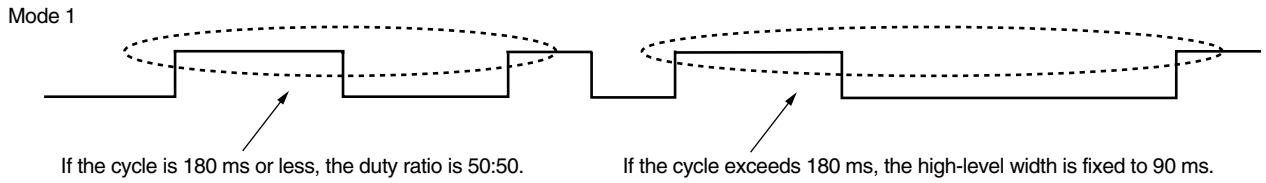
(a) Mode 1

Output in the range from 0 to 135 Hz is possible.

A waveform that has a pulse duty ratio of 50:50 is output.

However, if the cycle exceeds 180 ms, the high-level width is fixed to 90 ms.

Figure 21-9. Waveform Output



(b) mode 2

Output in the range from 0 to 1,085 Hz is possible.

A pulse waveform that has a fixed high-level width is output. The high-level width can be specified for the PULCTL register as 30, 60, 90, or 180 μs.

This mode is mainly used in Japan.

CHAPTER 22 STANDBY FUNCTION

22.1 Standby Function and Configuration

22.1.1 Standby function

The standby function is mounted onto all 78K0R/Lx3-M microcontroller products.

The standby function reduces the operating current of the system, and the following two 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, internal high-speed oscillator, 20 MHz internal high-speed 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 internal high-speed oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions

1. **The STOP mode can be used only when the CPU is operating on the main system clock. The STOP mode cannot be set 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.**
3. **The following sequence is recommended for operating current reduction of the 10-bit successive approximation type A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the STOP instruction. The following sequence is recommended for operating current reduction of the 24-bit ΔΣ-type A/D converter when the standby function is used: First clear bit 7 (ADPON) and bit 6 (ADCE2) of the 24-bit ΔΣ-type A/D converter mode register (ADM2) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.**
4. **It can be selected by the option byte whether the internal low-speed oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 27 OPTION BYTE.**
5. **The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.**

22.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For the registers that start, stop, or select the clock, see **CHAPTER 5 CLOCK GENERATOR**.

(1) Oscillation stabilization time counter status register (OSTC)

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

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

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

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

When reset is released (reset by $\overline{\text{RESET}}$ input, POC, LVI, WDT, and executing an illegal instruction), the STOP instruction and MSTOP (bit 7 of CSC register) = 1 clear this register to 00H.

Figure 22-1. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H After reset: 00H R

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

MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	102.4 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	204.8 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	819.2 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	26.21 ms min.	13.11 ms min.

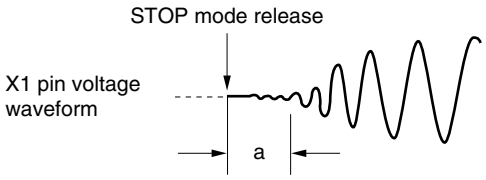
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

• Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

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



Remark fx: X1 clock oscillation frequency

(2) Oscillation stabilization time select register (OSTS)

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

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

When the internal high-speed oscillation clock is selected as the CPU clock, confirm with OSTC that the desired oscillation stabilization time has elapsed after the STOP mode is released. The oscillation stabilization time can be checked up to the time set using OSTC.

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

Reset signal generation sets this register to 07H.

Figure 22-2. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H After reset: 07H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	Setting prohibited
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_X$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_X$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.21 ms	13.11 ms

Cautions 1. To set the STOP mode when the X1 clock is used as the CPU clock, set OSTS before executing the STOP instruction.

2. Setting the oscillation stabilization time to 20 μs or less is prohibited.

3. Before changing the setting of the OSTS register, confirm that the count operation of the OSTC register is completed.

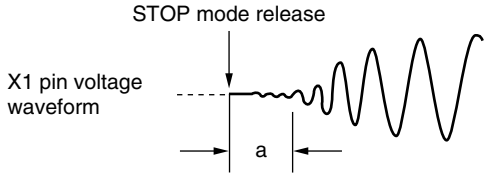
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

5. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

• Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

6. 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

22.2 Standby Function Operation

22.2.1 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, internal high-speed oscillation clock, 20 MHz internal high-speed oscillation clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Table 22-1. Operating Statuses in HALT Mode (1/3)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{IH}) or 20 MHz Internal High-Speed Oscillation Clock (f_{IH20})	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)	Status before HALT mode was set is retained	
		f_X	Status before HALT mode was set is retained	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 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Oscillates • WTON = 1 and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			Status before HALT mode was set is retained at voltage higher than POC detection voltage.		
Port (latch)			Status before HALT mode was set is retained		
Timer array unit (TAU)			Operable		
Real-time counter (RTC)					
Real-time counter 2 (RTC2)					
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Operates • WTON = 1 and WDSTBYON = 0: Stops		
Clock output			Operable		
10-bit successive approximation type A/D converter					
24-bit ΔΣ-type A/D converter					
Serial array unit (SAU)					
Serial interface (IICA)					
LCD controller/driver					
Multiplier/divider					
DMA controller					
Power-on-clear function					
Low-voltage detection function					
External interrupt					
Extended SFR (3rd SFR) interface					
Power calculation circuit					
Power quality measurement circuit					
Digital frequency conversion circuit					

Remarks 1. f_{IH}: Internal high-speed oscillation clock, f_{IH20}: 20 MHz internal high-speed oscillation clock
f_X: X1 oscillation clock, f_{EX}: External main system clock
f_{XT}: XT1 oscillation clock, f_{EXS}: External subsystem clock
f_{IL}: Internal low-speed oscillation clock

2. The functions mounted depend on the product. Refer to **1.4 Block Diagram** and **1.5 Outline of Functions**.

Table 22-1. Operating Statuses in HALT Mode (2/3)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock
			When CPU Is Operating on External Subsystem Clock (f_{EXS})
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{IH}	Status before HALT mode was set is retained
		f_X	
		f_{EX}	Operates or stops by external clock input
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained
		f_{EXS}	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Oscillates • WTON = 1 and WDSTBYON = 0: Stops
CPU			Operation stopped
Flash memory			Operation stopped (wait state in low-power consumption mode)
RAM			Status before HALT mode was set is retained at voltage higher than POC detection voltage.
Port (latch)			Status before HALT mode was set is retained
Timer array unit (TAU)			Operable
Real-time counter (RTC)			
Real-time counter 2 (RTC2)			
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Operates • WTON = 1 and WDSTBYON = 0: Stops
Clock output			Operable
10-bit successive approximation type A/D converter			Cannot operate
24-bit ΔΣ-type A/D converter			Operable
Serial array unit (SAU)			
Serial interface (IICA)			Cannot operate
LCD controller/driver			Operable
Multiplier/divider			
DMA controller			
Power-on-clear function			
Low-voltage detection function			
External interrupt			
Extended SFR (3rd SFR) interface			
Power calculation circuit			
Power quality measurement circuit			
Digital frequency conversion circuit			

Remarks 1. f_{IH}: Internal high-speed oscillation clock, f_X: X1 oscillation clock,
f_{EX}: External main system clock f_{XT}: XT1 oscillation clock,
f_{EXS}: External subsystem clock f_{IL}: Internal low-speed oscillation clock

2. The functions mounted depend on the product. Refer to **1.4 Block Diagram** and **1.5 Outline of Functions**.

Table 22-1. Operating Statuses in HALT Mode (3/3)

Item \ HALT Mode Setting			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock
			When CPU Is Operating on External Subsystem Clock (f_{EXS})) (Subsystem Clock HALT Mode (RTCLPC = 1))
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{IH}	Status before HALT mode was set is retained
		f_X	
		f_{EX}	Operates or stops by external clock input
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained
		f_{EXS}	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Oscillates • WTON = 1 and WDSTBYON = 0: Stops
CPU			Operation stopped
Flash memory			Operation stopped (wait state in low-power consumption mode)
RAM			Status before HALT mode was set is retained at voltage higher than POC detection voltage.
Port (latch)			Status before HALT mode was set is retained
Timer array unit (TAU)			Cannot operate
Real-time counter (RTC)			Operable
Real-time counter 2 (RTC2)			
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Operates • WTON = 1 and WDSTBYON = 0: Stops
Clock output			Operable
10-bit successive approximation type A/D converter			Cannot operate
24-bit ΔΣ-type A/D converter			Operable
Serial array unit (SAU)			Cannot operate
Serial interface (IICA)			
LCD controller/driver			Operable
Multiplier/divider			Operation stopped
DMA controller			
Power-on-clear function			Operable
Low-voltage detection function			
External interrupt			
Extended SFR (3rd SFR) interface			Cannot operate
Power calculation circuit			Operable
Power quality measurement circuit			
Digital frequency conversion circuit			

Remarks 1. f_{IH}: Internal high-speed oscillation clock, f_X: X1 oscillation clock,
f_{EX}: External main system clock f_{XT}: XT1 oscillation clock,
f_{EXS}: External subsystem clock f_{IL}: Internal low-speed oscillation clock

2. RTCLPC: Bit 7 of the operation speed mode control register (OSMC).

3. The functions mounted depend on the product. Refer to **1.4 Block Diagram** and **1.5 Outline of Functions**.

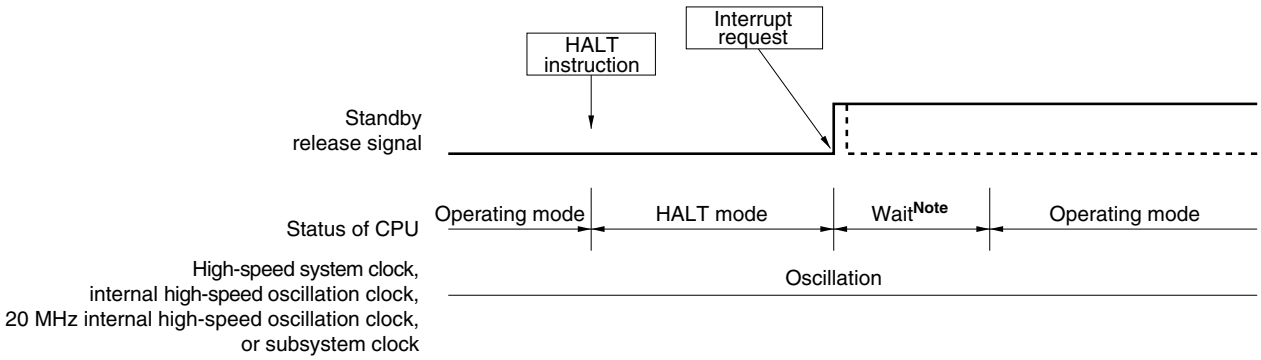
(2) HALT mode release

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 22-3. HALT Mode Release by Interrupt Request Generation



Note The wait time is as follows:

- When vectored interrupt servicing is carried out
  - When main system clock is used: 10 to 12 clocks
  - When subsystem clock is used: 8 to 10 clocks
- When vectored interrupt servicing is not carried out
  - When main system clock is used: 5 or 6 clocks
  - When subsystem clock is used: 3 or 4 clocks

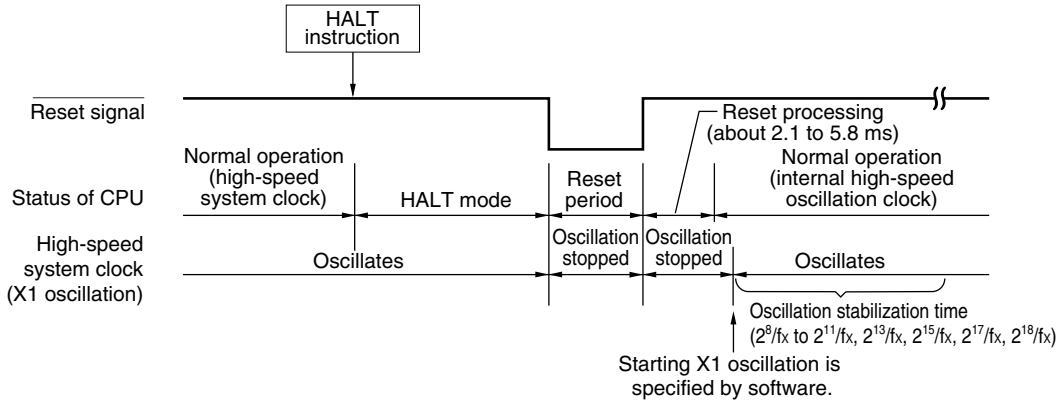
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

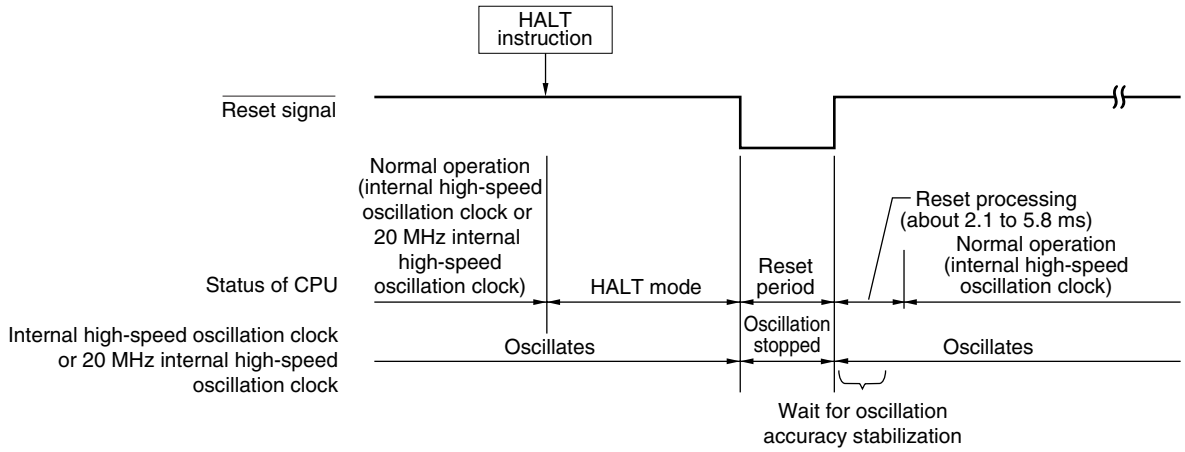
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 22-4. HALT Mode Release by Reset

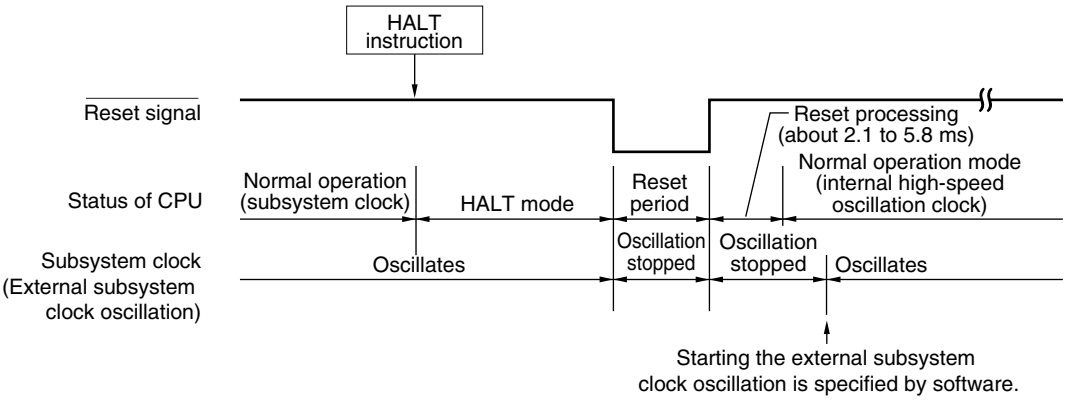
(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock or 20 MHz internal high-speed oscillation clock is used as CPU clock



(3) When subsystem clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

22.2.2 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 main system clock.

Cautions 1. Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

2. The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.

The operating statuses in the STOP mode are shown below.

Table 22-2. Operating Statuses in STOP Mode

Item \ STOP Mode Setting		When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
		When CPU Is Operating on Internal High-Speed Oscillation 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 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Oscillates • WTON = 1 and WDSTBYON = 0: Stops		
CPU		Operation stopped		
Flash memory				
RAM		Status before STOP mode was set is retained at voltage higher than POC detection voltage.		
Port (latch)		Status before STOP mode was set is retained		
Timer array unit (TAU)		Cannot operate		
Real-time counter (RTC)		Operable		
Real-time counter 2 (RTC2)				
Watchdog timer		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Operates • WTON = 1 and WDSTBYON = 0: Stops		
Clock output		Operable only when subsystem clock is selected as the count clock		
10-bit successive approximation type A/D converter		Operation stopped		
24-bit ΔΣ-type A/D converter				
Serial array unit (SAU)		Cannot operate		
Serial interface (IICA)		Wake-up by address match operable		
LCD controller/driver		Operable only when subsystem clock is selected as LCD source clock		
Multiplier/divider		Cannot operate		
DMA controller				
Power-on-clear function		Operable		
Low-voltage detection function				
External interrupt				
Extended SFR (3rd SFR) interface		Cannot operate		
Power calculation circuit				
Power quality measurement circuit				
Digital frequency conversion circuit				

Remarks 1. f_{IH}: Internal high-speed oscillation clock, f_X: X1 oscillation clock,
f_{EX}: External main system clock f_{XT}: XT1 oscillation clock,
f_{EXS}: External subsystem clock f_{IL}: Internal low-speed oscillation clock

2. The functions mounted depend on the product. Refer to **1.4 Block Diagram** and **1.5 Outline of Functions**.

(**Cautions** are listed on the next page.)

Cautions 1. To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.

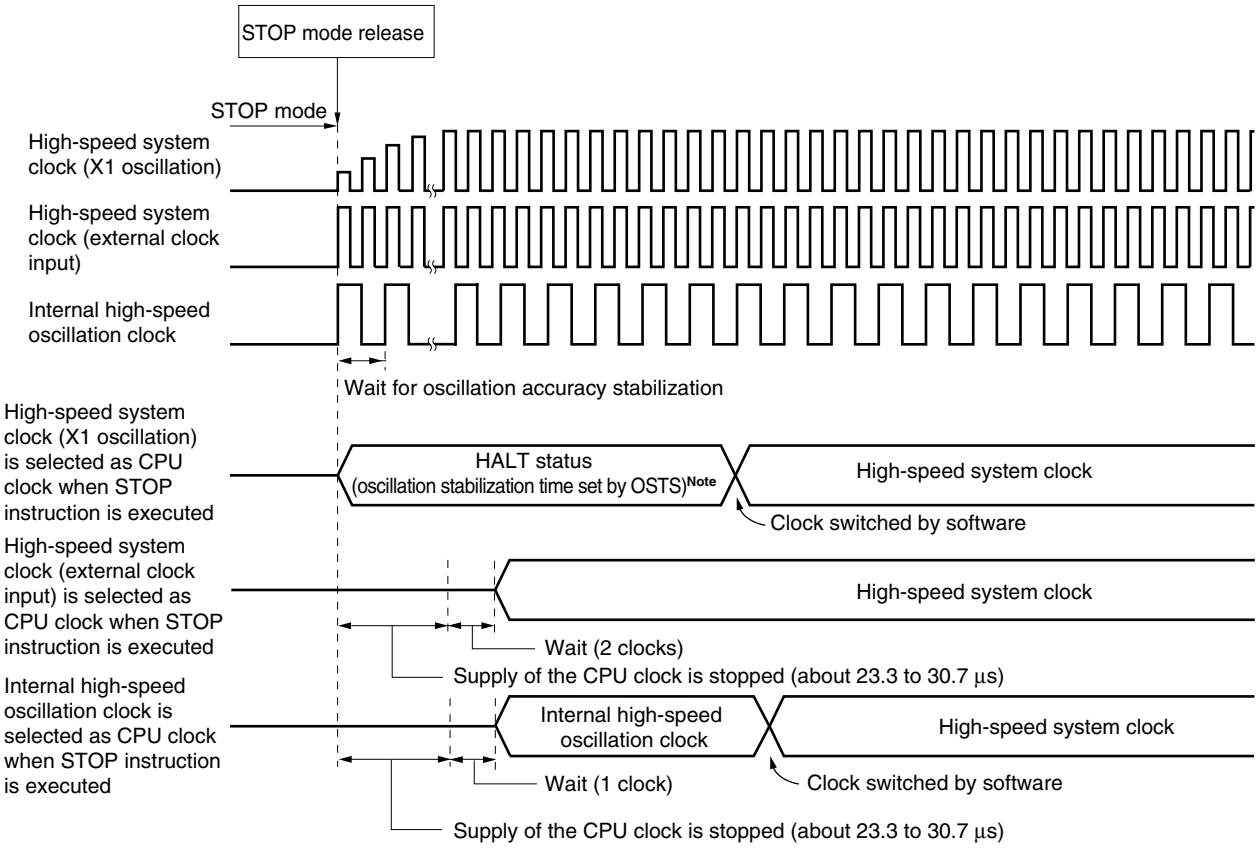
2. To stop the internal low-speed oscillation clock in the STOP mode, use an option byte to stop the watchdog timer operation in the HALT/STOP mode (bit 0 (WDSTBYON) of 000C0H = 0), and then execute the STOP instruction.

3. To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the internal high-speed oscillation clock before the next execution of the STOP instruction. Before changing the CPU clock from the internal high-speed oscillation clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).

4. The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.

(2) STOP mode release

Figure 22-5. Operation Timing When STOP Mode Is Released (When Unmasked Interrupt Request Is Generated)



Note When the oscillation stabilization time set by OSTS is equal to or shorter than 61 μs, the HALT status is retained to a maximum of "61μs + wait time."

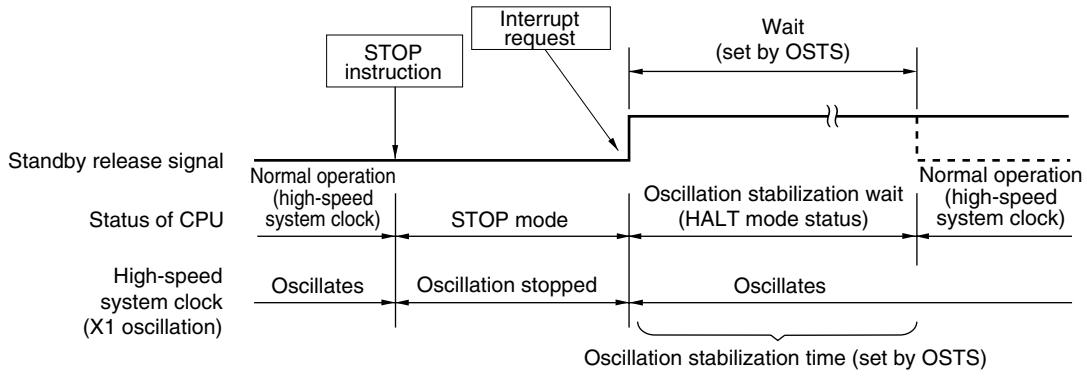
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

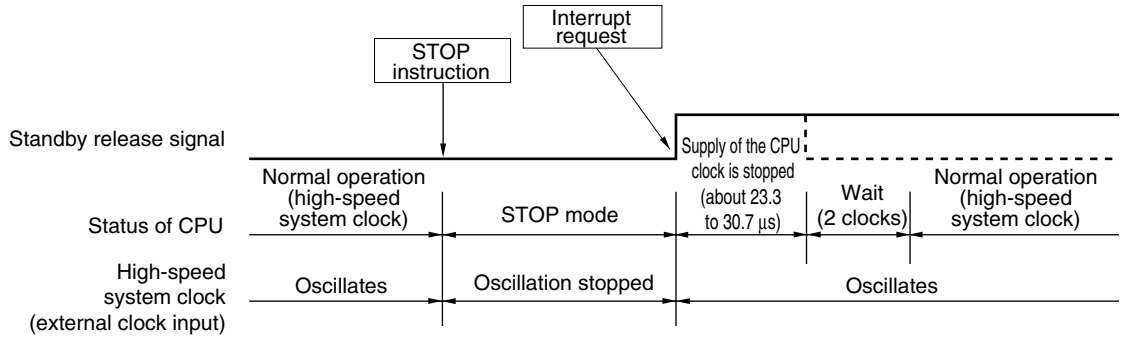
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 22-6. STOP Mode Release by Interrupt Request Generation

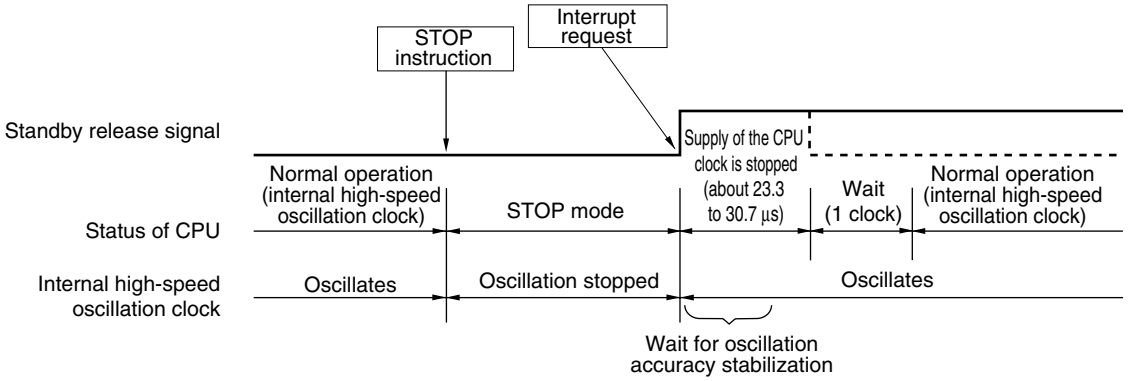
(1) When high-speed system clock (X1 oscillation) is used as CPU clock



(2) When high-speed system clock (external clock input) is used as CPU clock



(3) When internal high-speed oscillation clock is used as CPU clock



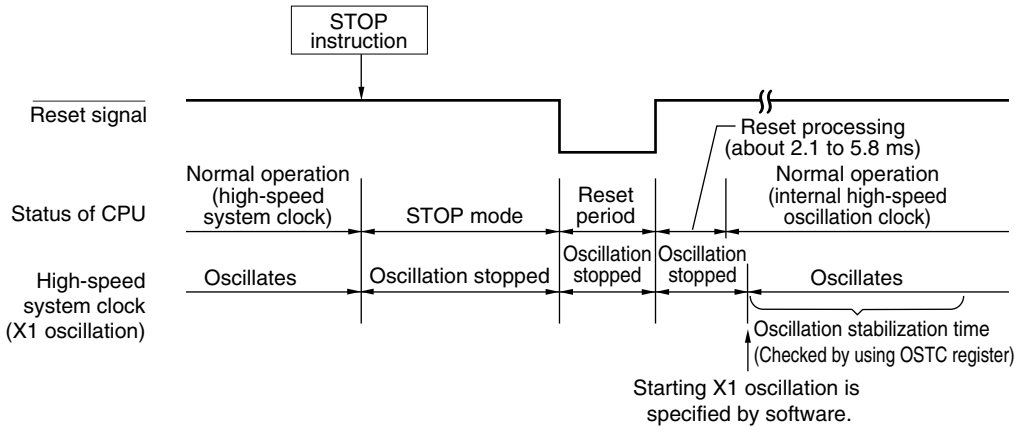
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

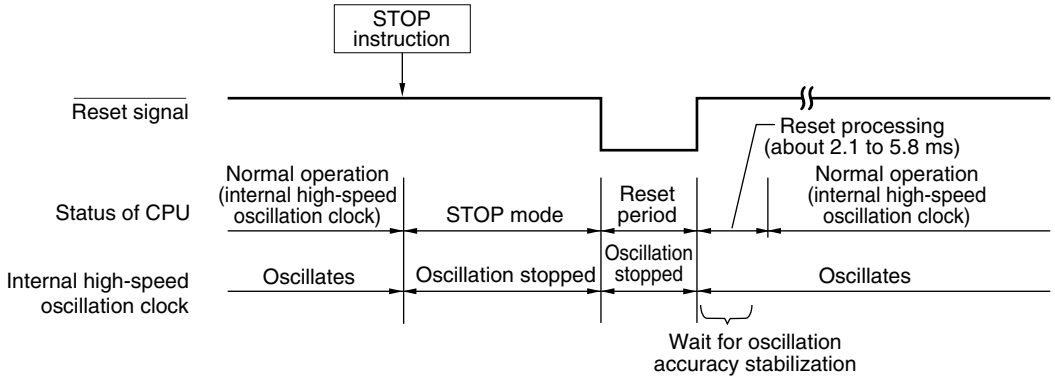
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 22-7. STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

CHAPTER 23 RESET FUNCTION

The following five operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-clear (POC) circuit
(4) Internal reset by comparison of supply voltage of the low-voltage detector (LVI) or input voltage (EXLVI) from external input pin, and detection voltage
(5) Internal reset by execution of illegal instruction[Note]

External and internal resets start program execution from the address at 0000H and 0001H 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 POC and LVI circuit voltage detection or execution of illegal instruction[Note], and each item of hardware is set to the status shown in Tables 23-1 and 23-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release.

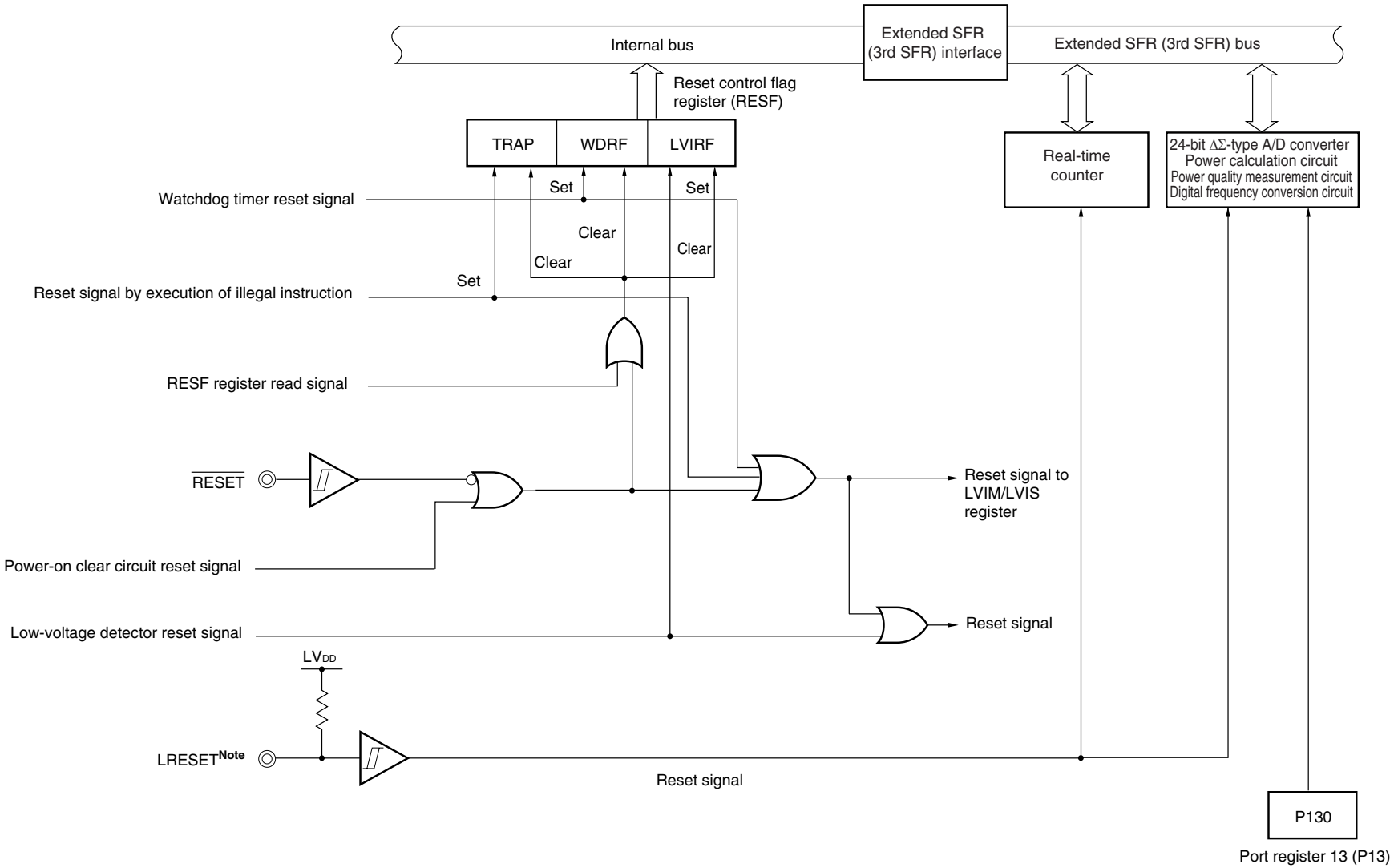
When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the internal high-speed oscillation clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the internal high-speed oscillation clock (see **Figures 23-2** to **23-4**) after reset processing. Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVI}$ after the reset, and program execution starts using the internal high-speed oscillation clock (see **CHAPTER 24 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 25 LOW-VOLTAGE DETECTOR**) after reset processing.

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

Cautions
1. **For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin**
   (To perform an external reset upon power application, a low level of at least 10 μs must be continued during the period in which the supply voltage is within the operating range ($V_{DD} \geq 1.8$ V)).
2. **During reset input, the X1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.**
3. **When the STOP mode is released by a reset, the RAM contents in the STOP mode are held during reset input.**
4. **When reset is effected, port pins become high-impedance, because each SFR and 2nd SFR are initialized.**
5. **The extended SFR (3rd SFR) is not reset by resetting. To reset the extended SFR (3rd SFR), clear P130 → set 1 → clear 0.**

Remark V_{POR}: POC power supply rise detection voltage
V_{LVI}: LVI detection voltage

Figure 23-1. Block Diagram of Reset Function



Note Be sure to connect it directly to LV_{DD} or via a resistor.

Caution An LVI circuit internal reset does not reset the LVI circuit.

Remarks 1. LVIM: Low-voltage detection register

2. LVIS: Low-voltage detection level select register

Figure 23-2. Timing of Reset by $\overline{\text{RESET}}$ Input

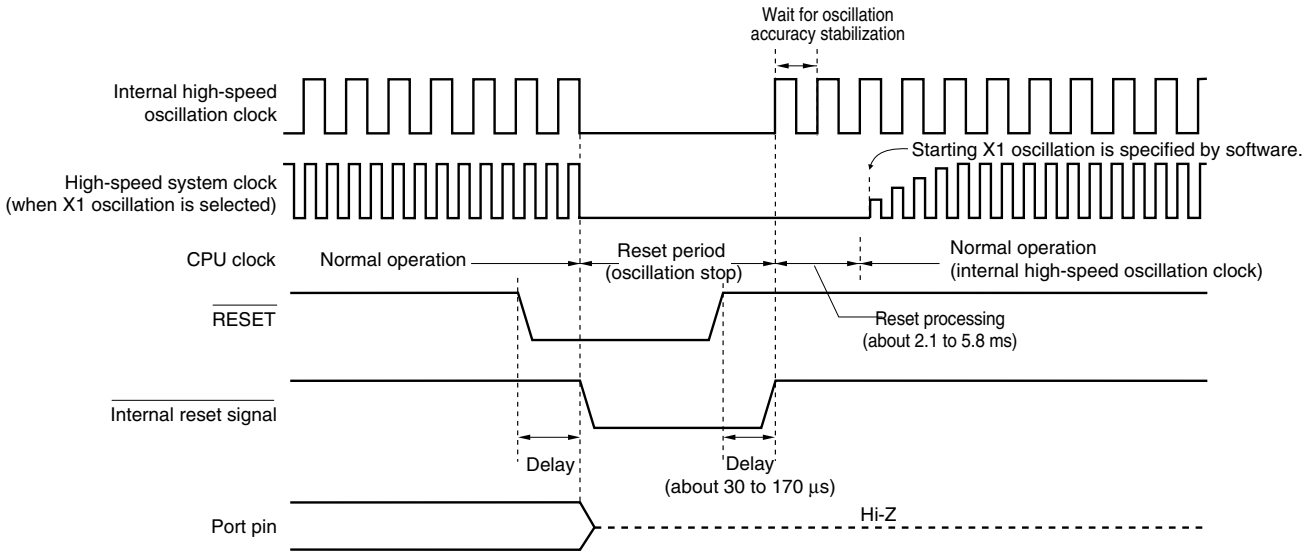
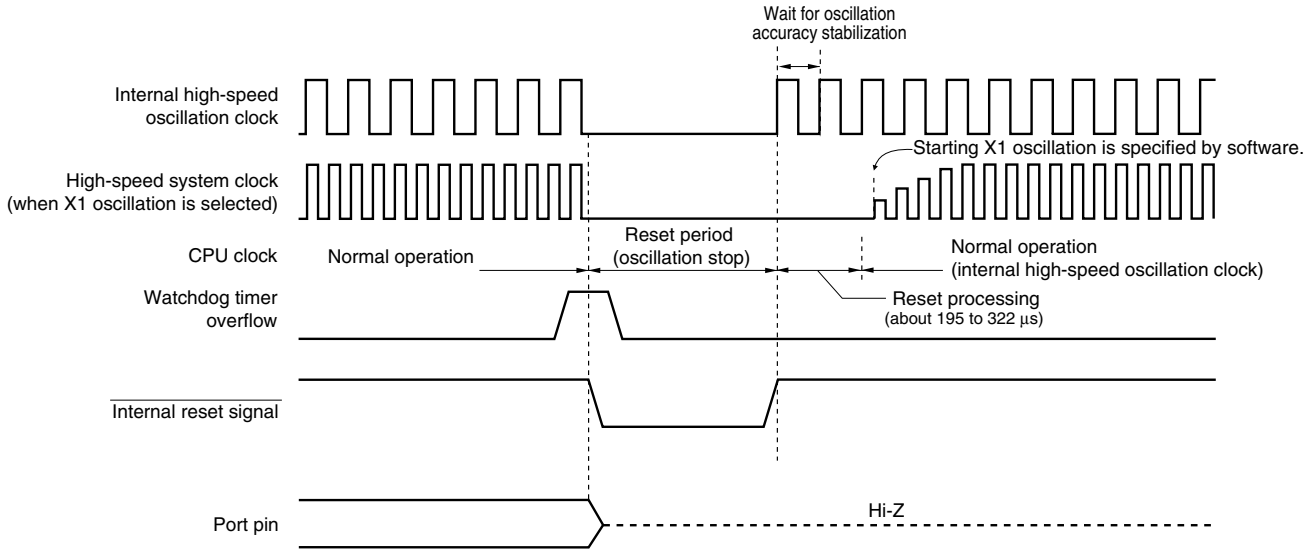
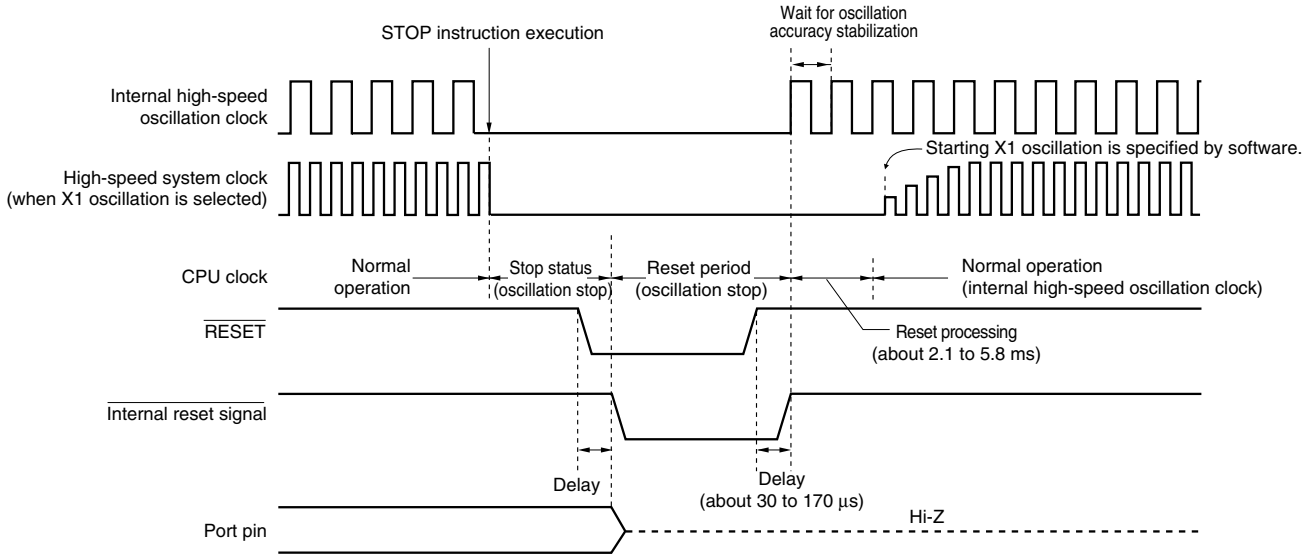


Figure 23-3. Timing of Reset Due to Watchdog Timer Overflow



Caution A watchdog timer internal reset resets the watchdog timer.

Figure 23-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Remark For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 24 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 25 LOW-VOLTAGE DETECTOR**.

Table 23-1. 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 (X1 and X2 pins are input port mode)
		f_{EX}	Clock input invalid (pin is input port mode)
	Subsystem clock	f_{XT}	Operable
		f_{XTS}	
	f_{IL}		Operation stopped
CPU			
Flash memory			
RAM			Operation stopped (The value, however, is retained when the voltage is at least the power-on-clear detection voltage.)
Port (latch)			High impedance.
Timer array unit (TAU)			Operation stopped
Real-time counter (RTC)			Operable
Real-time counter 2 (RTC2)			Operation stopped
Watchdog timer			
Clock output			
10-bit successive approximation type A/D converte			
24-bit ΔΣ-type A/D converter			
Serial array unit (SAU)			
Serial interface (IICA)			
LCD controller/driver			Operation stopped (COM only pin, SEG only pin: GND output, SEG/general-purpose port alternate pin: input port, V_{LC0} to V_{LC2} pins: high-impedance output, CAPH/P00 pin, CAPL/P01 pin: input port)
Multiplier/divider			Operation stopped
DMA controller			
Power-on-clear function			Detection operation possible
Low-voltage detection function			Operation stopped (however, operation continues at LVI reset)
External interrupt			Operation stopped
BCD correction circuit (BCD)			
Extended SFR (3rd SFR) interface			
Power calculation circuit			
Power quality measurement circuit			
Digital frequency conversion circuit			

Remarks 1. f_{IH}: Internal high-speed oscillation clock, f_X: X1 oscillation clock
f_{EX}: External main system clock, f_{XT}: XT1 oscillation clock
f_{EXS}: External subsystem clock, f_{IL}: Internal low-speed oscillation clock

2. The functions mounted depend on the product. Refer to **1.4 Block Diagram** and **1.5 Outline of Functions**.

Table 23-2. Hardware Statuses After Reset Acknowledgment (1/6)

Hardware		After Reset Acknowledgment[Note 1]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Port registers (P0 to P6, P8 to P10, P12 to P15, LP0) (output latches)		00H
Port mode registers (PM0 to PM6, PM8 to PM15, LPM0)		FFH
Port input mode register 1 (PIM1)		00H
Port output mode registers 1, 8 (POM1, POM8)		00H
Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU8 to PU10, PU12, PU14, LPU0)		00H
Clock operation mode control register (CMC)		00H
Por function control register (PORTCTL)		00H
Clock operation status control register (CSC)		C0H
Processor mode control register (PMC)		00H
System clock control register (CKC)		09H
20 MHz internal high-speed oscillation control register (DSCCTL)		00H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		07H
Noise filter enable registers 0, 1 (NFEN0, NFEN1)		00H
Peripheral enable register 0 (PER0)		00H
Operation speed mode control register (OSMC)		00H
Input switch control register (ISC)		00H
Timer array units 0, 1 (TAU0, TAU1)	Timer data registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13 (TDR00, TDR01, TDR02, TDR03, TDR04, TDR05, TDR06, TDR07, TDR10, TDR11, TDR12, TDR13)	0000H
	Timer mode registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13 (TMR00, TMR01, TMR02, TMR03, TMR04, TMR05, TMR06, TMR07, TMR10, TMR11, TMR12, TMR13)	0000H
	Timer status registers 02, 04, 07 (TSR02, TSR04, TSR07)	0000H
	Timer input select registers 0, 1 (TIS0, TIS1)	00H
	Timer channel counter registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13 (TCR00, TCR01, TCR02, TCR03, TCR04, TCR05, TCR06, TCR07, TCR10, TCR11, TCR12, TCR13)	FFFFH
	Timer channel enable status registers 0, 1 (TE0, TE1)	0000H
	Timer channel start trigger registers 0, 1 (TS0, TS1)	0000H
	Timer channel stop trigger registers 0, 1 (TT0, TT1)	0000H
	Timer clock select registers 0, 1 (TPS0, TPS1)	0000H
	Timer channel output register 0 (TO0)	0000H
	Timer channel output enable register 0 (TOE0)	0000H

Notes 1. 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.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

Table 23-2. Hardware Statuses After Reset Acknowledgment (2/6)

Hardware		Status After Reset Acknowledgment[Note 1]
Timer array units 0, 1 (TAU0, TAU1)	Timer channel output level register 0 (TOL0)	0000H
	Timer channel output mode register 0 (TOM0)	0000H
Real-time counter 2	Sub-count register (RSUBC)	0000H
	Second count register (SEC)	00H
	Minute count register (MIN)	00H
	Hour count register (HOUR)	12H
	Week count register (WEEK)	00H
	Day count register (DAY)	01H
	Month count register (MONTH)	01H
	Year count register (YEAR)	00H
	Watch error correction register (SUBCUD)	00H
	Alarm minute register (ALARMWM)	00H
	Alarm hour register (ALARMWH)	12H
	Alarm week register (ALARMWW)	00H
	Real-time counter control register 0 (RTCC0)	00H
	Real-time counter control register 1 (RTCC1)	00H
	Real-time counter control register 2 (RTCC2)	00H
Clock output controller	Clock output select register 0 (CKS0)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 2]
10-bit successive approximation type A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	A/D converter mode register (ADM)	00H
	A/D converter mode register 1 (ADM1)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	10H
24-bit ΔΣ-type A/D converter	24-bit ΔΣ-type A/D converter mode register (ADM2)	00H
	High-pass filter control register 0 (HPFC0)	00H
	High-pass filter control register 1 (HPFC1)	00H
	24-bit ΔΣ-type A/D conversion result register (ADCR0, ADCR1, ADCR2, ADCR3)	000000H
	Phase control registers 0 and 1 (PHC0, PHC1)	0000H
	A/D clock delay setting register (ADLY)	00H

Notes 1. 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.

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

Table 23-2. Hardware Statuses After Reset Acknowledgment (3/6)

Hardware		Status After Reset Acknowledgment[Note]
Serial array units 0, 1 (SAU0, SAU1)	Serial data registers 00, 01, 10, 11, 12, 13 (SDR00, SDR01, SDR10, SDR11, SDR12, SDR13)	0000H
	Serial status registers 00, 01, 10, 11, 12, 13 (SSR00, SSR01, SSR10, SSR11, SSR12, SSR13)	0000H
	Serial flag clear trigger registers 01, 10, 11, 13 (SIR01, SIR10, SIR11, SIR13)	0000H
	Serial mode registers 00, 01, 10, 11, 12, 13 (SMR00, SMR01, SMR10, SMR11, SMR12, SMR13)	0020H
	Serial communication operation setting registers 00, 01, 10, 11, 12, 13 (SCR00, SCR01, SCR10, SCR11, SCR12, SCR13)	0087H
	Serial channel enable status registers 0, 1 (SE0, SE1)	0000H
	Serial channel start registers 0, 1 (SS0, SS1)	0000H
	Serial channel stop registers 0, 1 (ST0, ST1)	0000H
	Serial clock select registers 0, 1 (SPS0, SPS1)	0000H
	Serial output registers 0, 1 (SO0, SO1)	0F0FH
	Serial output enable registers 0, 1 (SOE0, SOE1)	0000H
	Serial output level registers 0, 1 (SOL0, SOL1)	0000H
Serial interface IICA	Shift register (IICA)	00H
	Control register 0 (IICCTL0)	00H
	Control register 1 (IICCTL1)	00H
	Slave address register (SVA)	00H
	IICA low-level width setting register 0 (IICWL)	FFH
	IICA high-level width setting register 0 (IICWH)	FFH
	Status register (IICS)	00H
	Flag register (IICF)	00H
Extended SFR (3rd SFR) interface	Serial data register 02 (SDR02)	0000H
	Serial status register 02 (SSR02)	0000H
	Serial flag clear trigger register 02 (SIR02)	0000H
	Serial mode register 02 (SMR02)	0020H
	Serial communication operation setting register 02 (SCR02)	0087H
	Serial channel enable status register 0 (SE0)	0000H
	Serial channel start register 0 (SS0)	0000H
	Serial channel stop register 0 (ST0)	0000H
	Serial clock select register 0 (SPS0)	0000H
	Serial output register 0 (SO0)	0F0FH
	Serial output enable register 0 (SOE0)	0000H
	Serial output level register 0 (SOL0)	0000H

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.

Table 23-2. Hardware Statuses After Reset Acknowledgment (4/6)

Hardware		Status After Reset Acknowledgment[Note]
LCD controller/driver	LCD mode register (LCDMD)	00H
	LCD display mode register (LCDM)	00H
	LCD clock control register 0 (LCDC0)	00H
	LCD boost level control register (VLCD)	0FH
	Port function register (PFALL)	00H
	Segment enable register (SEGEN)	00H
	Input switch control register (ISC)	00H
Multiplier/divider	Multiplication/division data register A (MDAL, MDAH)	0000H
	Multiplication/division data register B (MDBL, MDBH)	0000H
	Multiplication/division data register C (MDCL, MDCH)	0000H
	Multiplication/division control register (MDUC)	00H
Power calculation circuit	Power calculation mode control register 1 (PWCTL1)	00H
	Power calculation mode control register 2 (PWCTL2)	00H
	No-load level control register (NLCTL)	00H
	Active power scaling specification register (ACTDIV)	00H
	Reactive power scaling specification register (READIV)	00H
	Apparent power scaling specification register (APPDIV)	00H
	RMS registers for voltage channels 1 and 2 (V1RMS, V2RMS)	000000H
	RMS registers for current channels 1 and 2 (I1RMS, I2RMS)	000000H
	Active power accumulation reading register (ACTHR)	000000H
	Active power accumulation reading and resetting register (RACTHR)	000000H
	Active power accumulation synchronous reading register (LACTHR)	000000H
	Reactive power accumulation reading register (REAHR)	000000H
	Reactive power accumulation reading and resetting register (RREAHR)	000000H
	Reactive power accumulation synchronous reading register (LREAHR)	000000H
	Apparent power accumulation reading register (APPHR)	000000H
	Apparent power accumulation reading and resetting register (RAPPHR)	000000H
	Apparent power accumulation synchronous reading register (LAPPHR)	000000H
	Line cycle number specification register (LINNUM)	FFFFH
	Active power gain specification registers 1 and 2 (ACT1GAIN, ACT2GAIN)	0000H
	Reactive power gain specification registers 1 and 2 (REA1GAIN, REA2GAIN)	0000H
	Apparent power gain specification registers 1 and 2 APP1GAIN, APP2GAIN)	0000H
	RMS gain specification registers for current channels 1 and 2 (IRMS1GAIN, IRMS2GAIN)	0000H
	Active power offset specification registers 1 and 2 (ACT1OS, ACT2OS)	0000H
	Reactive power offset specification registers 1 and 2 (REA1OS, REA2OS)	0000H
	RMS offset specification registers for current channels 1 and 2 (I1RMSOS, I2RMSOS)	0000H
	RMS offset specification registers for voltage channels 1 and 2 (V1RMSOS, V2RMSOS)	0000H
	Sampling mode selection register (SAMPMODE)	00H
	Sampling result registers 1 and 2 (SAMP1, SAMP2)	000000H

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.

Table 23-2. Hardware Statuses After Reset Acknowledgment (5/6)

Hardware		Status After Reset Acknowledgment[Note 1]
Power quality measurement circuit	Period and frequency measurement result register (PFVAL)	0000H
	Zero-crossing timeout specification registers for voltage channels 1 and 2 (ZXTOUT1, ZXTOUT2)	03FFH
	SAG line cycle number specification registers for voltage channels 1 and 2 (SAGNUM1, SAGNUM2)	FFH
	SAG level specification registers for voltage channels 1 and 2 (SAGVAL1, SAGVAL2)	000000H
	Peak current level specification register (IPKLMT)	FFFFH
	Peak voltage level specification register (VPKLMT)	FFFFH
	Peak current value register (IMAX)	000000H
	Peak current value clearing register (RSTIMAX)	000000H
	Peak voltage value register (VMAX)	000000H
	Peak voltage value clearing register (RSTVMAX)	000000H
	Gain specification register (IMATGAIN)	0000H
	Fault detection control register (PQMCTL)	00H
	Fault detection threshold value specification register (IST)	8BH
	Fault control register (ICHK)	84H
Digital frequency conversion circuit	Frequency conversion control register (CFCTL)	02H
	Frequency scaling specification register (CFMULL, CFMULH)	FFH
	Pulse width specification register (PULCTL)	00H
Reset function	Reset control flag register (RESF)	Undefined[Note 2]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note 3]
	Low-voltage detection level select register (LVIS)	0EH[Note 2]

Notes 1. 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.

2. These values vary depending on the reset source.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by LVI
RESF	TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held
	WDRF bit			Held	Set (1)	Held
	LVIRF bit			Held	Held	Set (1)
LVIS		Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Held

3. This value varies depending on the reset source and the option byte.

Table 23-2. Hardware Statuses After Reset Acknowledgment (6/6)

Hardware		Status After Reset Acknowledgment[Note 1]
DMA controller	SFR address registers 0, 1 (DSA0, DSA1)	00H
	RAM address registers 0L, 0H, 1L, 1H (DRA0L, DRA0H, DRA1L, DRA1H)	00H
	Byte count registers 0L, 0H, 1L, 1H (DBC0L, DBC0H, DBC1L, DBC1H)	00H
	Mode control registers 0, 1 (DMC0, DMC1)	00H
	Operation control registers 0, 1 (DRC0, DRC1)	00H
Interrupt	Request flag registers 0L, 0H, 1L, 1H, 2L, 2H (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)	00H
	Mask flag registers 0L, 0H, 1L, 1H, 2L, 2H (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)	FFH
	Extended SFR (3rd SFR) interrupt request flag registers 20, 21, 22, 23 (IF20, IF21, IF22, IF23)	00H
	Extended SFR (3rd SFR) Interrupt mask flag registers 20, 21, 22, 23 (MK20, MK21, MK22, MK23)	FFH
	Priority specification flag registers 00L, 00H, 01L, 01H, 02L, 02H, 10L, 10H, 11L, 11H, 12L, 12H (PR00L, PR00H, PR01L, PR01H, PR10L, PR10H, PR11L, PR11H, PR02L, PR02H, PR12L, PR12H)	FFH
	External interrupt rising edge enable registers 0, 1 (EGP0, EGP1)	00H
	External interrupt falling edge enable registers 0, 1 (EGN0, EGN1)	00H
Regulator	Regulator mode control register (RMC)	00H
Flash memory	Background event control register (BECTL)	00H
BCD correction circuit (BCD)	BCD correction result register (BCDADJ)	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.

23.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0R/Lx3-M microcontrollers. The reset control flag register (RESF) is used to store which source has generated the reset request.

RESF can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, reset by power-on-clear (POC) circuit, and reading RESF clear TRAP, WDRF, and LVIRF.

Figure 23-5. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP Note 1	Undefined	Undefined	WDRF Note 1	Undefined	Undefined	Undefined	LVIRF Note 1

TRAP	Internal reset request by execution of illegal instruction Note 2
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

WDRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by low-voltage detector (LVI)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

Notes 1. The value after reset varies depending on the reset source.

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

Cautions 1. Do not read data by a 1-bit memory manipulation instruction.

2. Do not make a judgment based on only the read value of the RESF register 8-bit data, because bits other than TRAP, WDRF, and LVIRF become undefined.

3. When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, LVIRF flag may become 1 from the beginning depending on the power-on waveform.

The status of RESF when a reset request is generated is shown in Table 23-3.

Table 23-3. RESF Status When Reset Request Is Generated

Flag \ Reset Source	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by LVI
TRAP	Cleared (0)	Cleared (0)	Set (1)	Held	Held
WDRF			Held	Set (1)	Held
LVIRF			Held	Held	Set (1)

23.2 Cautions after resetting

Be sure to set the following after resetting.

<1> Be sure to clear bits PM02, PM21 to PM27, PM80, PM110, PM111, PM150 and PM151 to 0 after reset release.

<2> The extended SFR (3rd SFR) is not reset by internal resetting. To reset the extended SFR (3rd SFR), clear P130
<R> → set 1 → clear 0. The minimum high-level width required for reset operation is 1 μsec (1000 nsec).

<3> Set PM32 (bit 2 of PM3 register) to the output enable (PM32 = 0), and select the output clock in clock output select register 0 (CKS0).

<R> <4> Set PM31, PM30, PM15, PM13 to the output enable (PM31, PM30, PM15, PM13 = 0), and set PM14 to input enable (PM14 = 1).

<5> Release the pull-down status (set 01H in PUTCTL register).

<6> When using P16/TO05, be sure to clear PM20 bit of PM2 to 0 after reset release, and to set P20 bit of P2 to 1 after reset release.

Caution If the internal resetting occurs during the data transmission in the extended SFR (3rdSFR) interface, it is possible that the data is not sent properly to the extended SFR (3rdSFR).

CHAPTER 24 POWER-ON-CLEAR CIRCUIT

24.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) is mounted onto all 78K0R/Lx3-M microcontroller products.
The power-on-clear circuit has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage (V_{DD}) exceeds 1.61 V ±0.09 V.

  Caution If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage (V_{DD}) exceeds 2.07 V ±0.2 V.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 V), generates internal reset signal when $V_{DD} < V_{PDR}$.

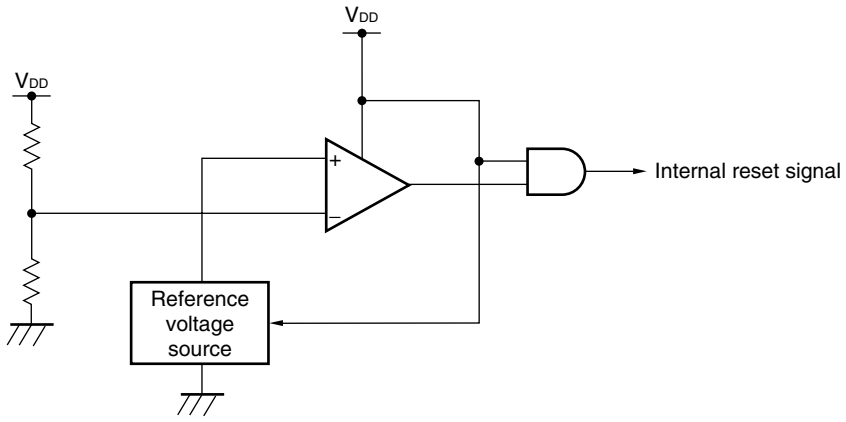
  Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

  Remark This product incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), low-voltage-detector (LVI), or illegal instruction execution. RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT or LVI.
  For details of RESF, see **CHAPTER 23 RESET FUNCTION**.

24.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 24-1.

Figure 24-1. Block Diagram of Power-on-Clear Circuit



24.3 Operation of Power-on-Clear Circuit

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{PDR} = 1.61 V ±0.09 V), the reset status is released.

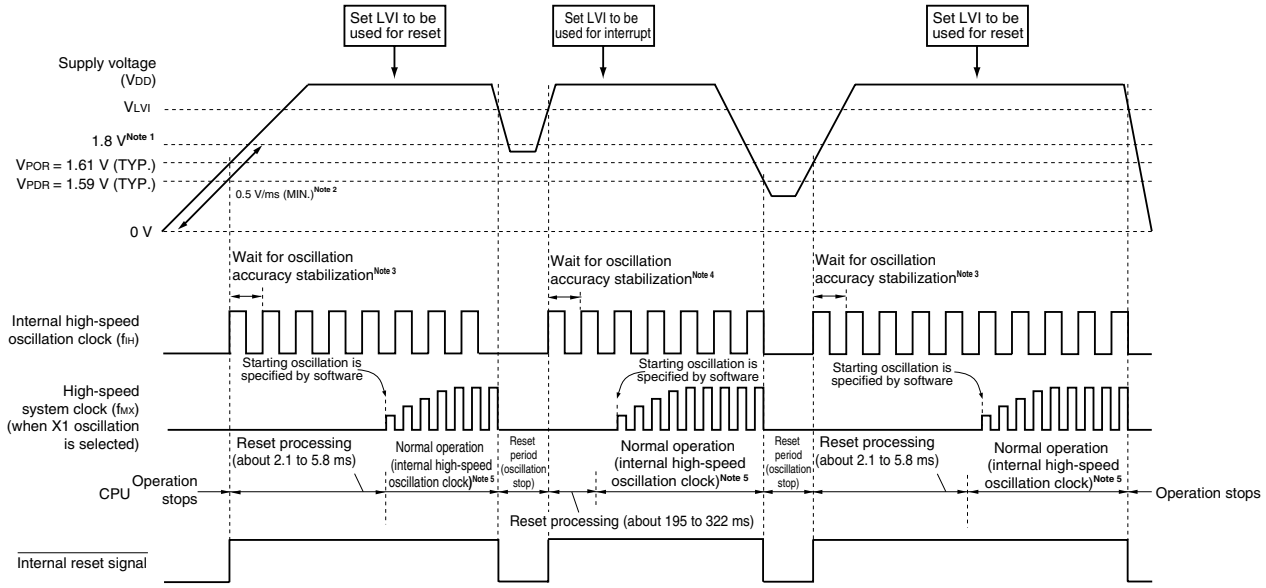
  Caution If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage (V_{DD}) exceeds 2.07 V ±0.2 V.

- The supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 V) are compared. When $V_{DD} < V_{PDR}$, the internal reset signal is generated.

The timing of generation of the internal reset signal by the power-on-clear circuit and low-voltage detector is shown below.

Figure 24-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)

(1) When LVI is OFF upon power application (option byte: LVIOFF = 1)



Notes 1. The operation guaranteed range is 1.8 V ≤ V_{DD} ≤ 3.6 V. To make the state at lower than 1.8 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. If the rate at which the voltage rises to 1.8 V after power application is slower than 0.5 V/ms (MIN.), input a low level to the $\overline{\text{RESET}}$ pin before the voltage reaches to 1.8 V, or set LVI to ON by default by using an option byte (option byte: LVIOFF = 0).

3. The reset processing time, such as when waiting for internal voltage stabilization, includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

4. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

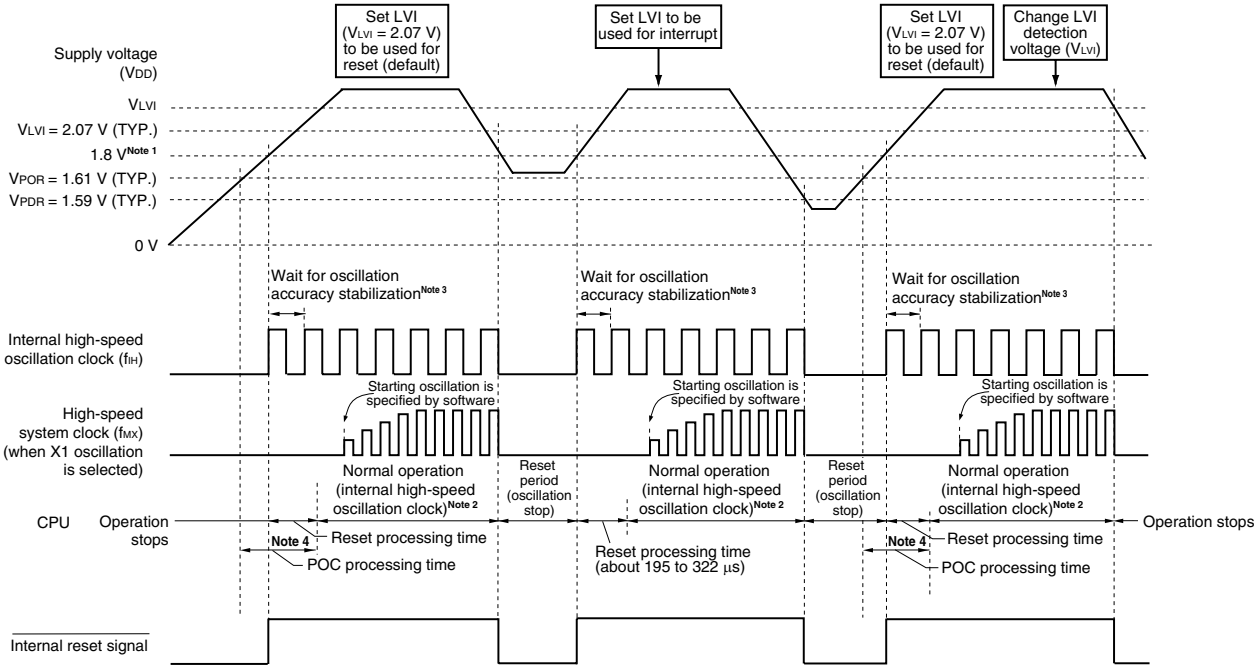
5. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 25 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

Figure 24-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)

(2) When LVI is ON upon power application (option byte: LVIOFF = 0)



Notes 1. The operation guaranteed range is 1.8 V ≤ V_{DD} ≤ 3.6 V. To make the state at lower than 1.8 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register 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 internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

4. The following times are required between reaching the POC detection voltage (1.59 V (TYP.)) and starting normal operation.

- When the time to reach 2.07 V (TYP.) from 1.59 V (TYP.) is less than 5.8 ms:
  A POC processing time of about 2.1 to 6.2 ms is required between reaching 1.59 V (TYP.) and starting normal operation.
- When the time to reach 2.07 V (TYP.) from 1.59 V (TYP.) is greater than 5.8 ms:
  A reset processing time of about 195 to 322 μs is required between reaching 2.07 V (TYP.) and starting normal operation.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 25 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

24.4 Cautions for Power-on-Clear Circuit

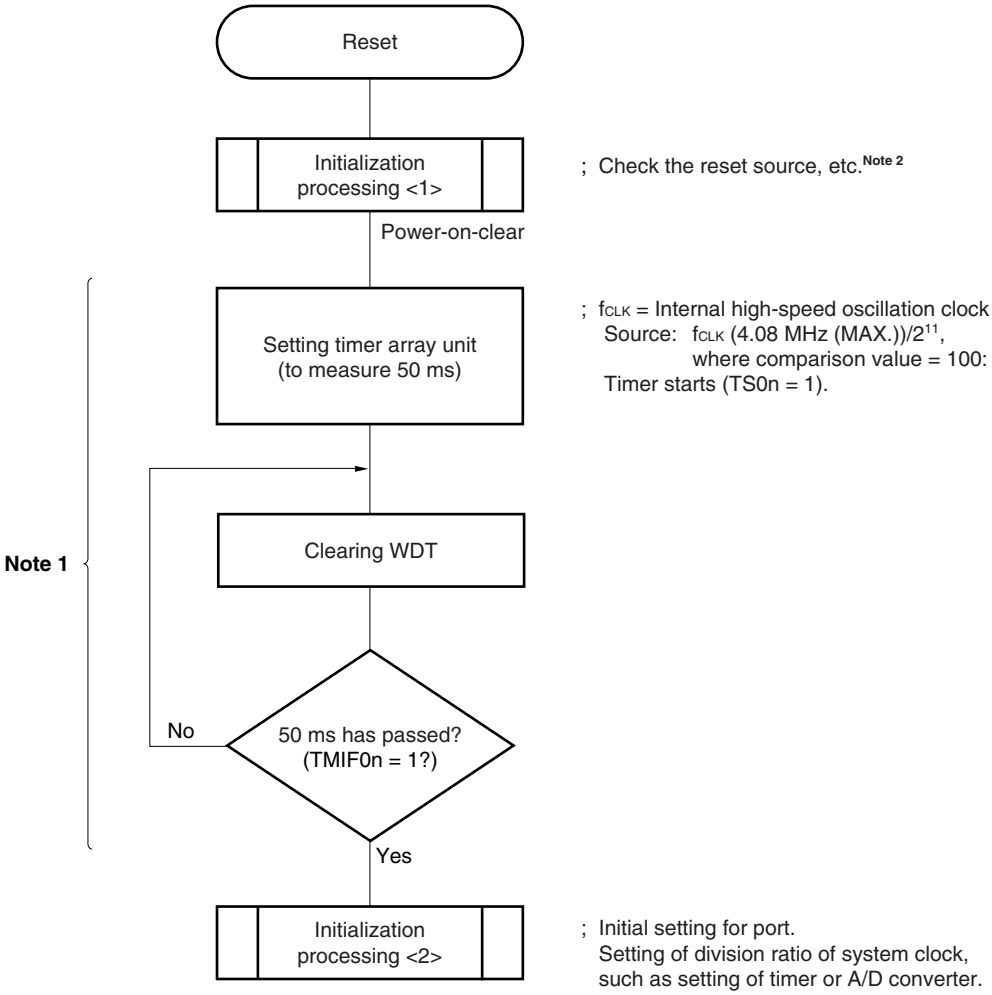
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POR}, V_{PDR}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller 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 24-3. Example of Software Processing After Reset Release (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage



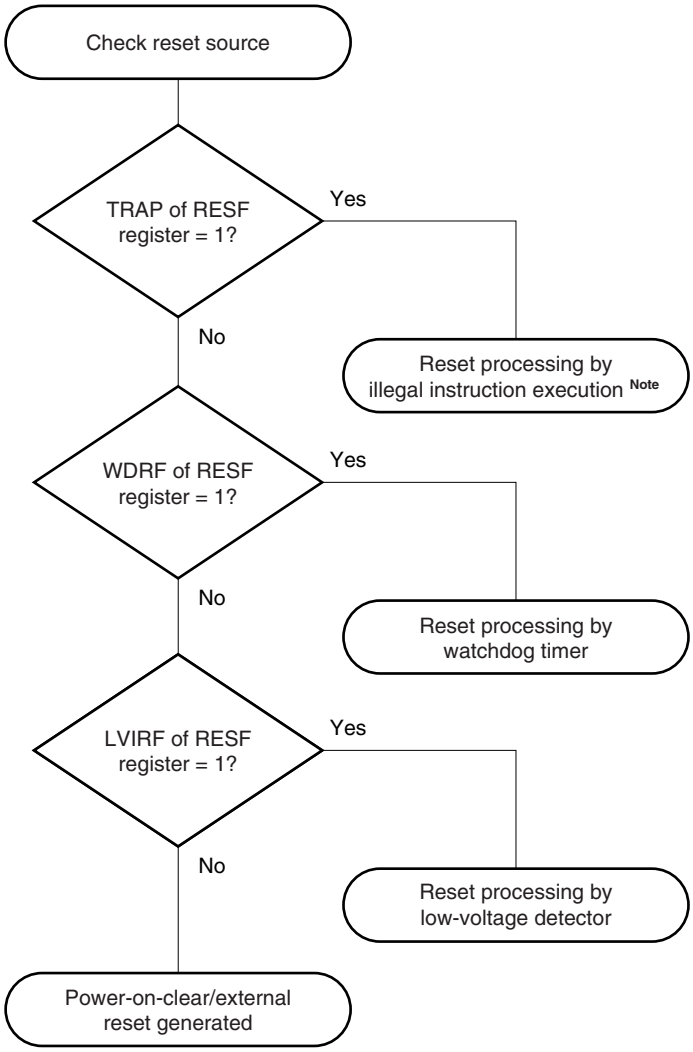
Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Remark n = 0 to 7

Figure 24-3. Example of Software Processing After Reset Release (2/2)

• Checking reset source



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

CHAPTER 25 LOW-VOLTAGE DETECTOR

25.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) is mounted onto all 78K0R/Lx3-M microcontroller products.
The low-voltage detector has the following functions.

- The LVI circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVI}) or the input voltage from an external input pin (EXLVI) with the detection voltage (V_{EXLVI} = 1.21 V ±0.1 V), and generates an internal reset or internal interrupt signal.
- The low-voltage detector (LVI) can be set to ON by an option byte by default. If it is set to ON to raise the power supply from the POC detection voltage (V_{POR} = 1.61 V (TYP.)) or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.2 V). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 V).
- The supply voltage (V_{DD}) or the input voltage from the external input pin (EXLVI) can be selected to be detected by software.
- A reset or an interrupt can be selected to be generated after detection by software.
- Detection levels (V_{LVI}, 11 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

The reset and interrupt signals are generated as follows depending on selection by software.

Selection of Level Detection of Supply Voltage (V_{DD}) (LVISEL = 0)		Selection Level Detection of Input Voltage from External Input Pin (EXLVI) (LVISEL = 1)	
Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).	Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).
Generates an internal reset signal when $V_{DD} < V_{LVI}$ and releases the reset signal when $V_{DD} \geq V_{LVI}$.	Generates an internal interrupt signal when V_{DD} drops lower than V_{LVI} ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$).	Generates an internal reset signal when EXLVI < V_{EXLVI} and releases the reset signal when EXLVI ≥ V_{EXLVI}.	Generates an internal interrupt signal when EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}).

Remark LVISEL: Bit 2 of low-voltage detection register (LVIM)
LVIMD: Bit 1 of LVIM

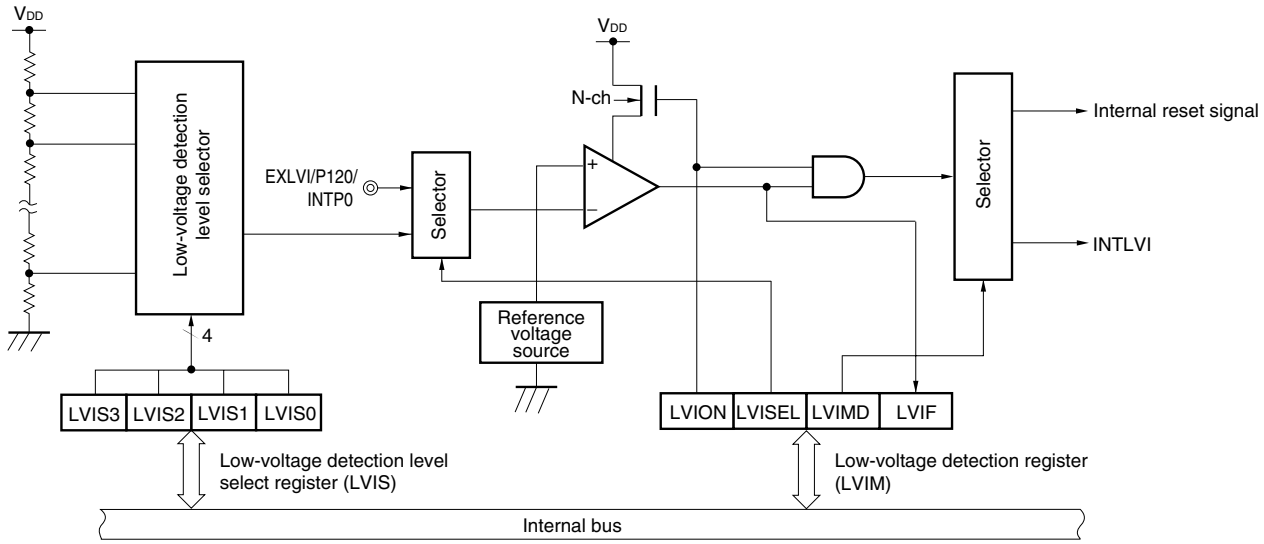
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

When the low-voltage detector is used to reset, bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of RESF, see **CHAPTER 23 RESET FUNCTION**.

25.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 25-1.

Figure 25-1. Block Diagram of Low-Voltage Detector



25.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection register (LVIM)
- Low-voltage detection level select register (LVIS)
- Port mode register 12 (PM12)

(1) Low-voltage detection register (LVIM)

This register sets low-voltage detection and the operation mode.

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

Reset signal generation clears this register to 00H.

Figure 25-2. Format of Low-Voltage Detection Register (LVIM)

Address: FFFA9H After reset: 00H[Note 1] R/W[Note 2]

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
LVIM	LVION	0	0	0	0	LVISEL	LVIMD	LVIF

LVION[Notes 3, 4]	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVISEL[Note 3]	Voltage detection selection
0	Detects level of supply voltage (V_{DD})
1	Detects level of input voltage from external input pin (EXLVI)

LVIMD[Note 3]	Low-voltage detection operation mode (interrupt/reset) selection
0	• LVISEL = 0: Generates an internal interrupt signal when the supply voltage (V_{DD}) drops lower than the detection voltage (V_{LVI}) ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$). • LVISEL = 1: Generates an interrupt signal when the input voltage from an external input pin (EXLVI) drops lower than the detection voltage (V_{EXLVI}) (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}).
1	• LVISEL = 0: Generates an internal reset signal when the supply voltage (V_{DD}) < detection voltage (V_{LVI}) and releases the reset signal when $V_{DD} \geq V_{LVI}$. • LVISEL = 1: Generates an internal reset signal when the input voltage from an external input pin (EXLVI) < detection voltage (V_{EXLVI}) and releases the reset signal when EXLVI ≥ V_{EXLVI}.

LVIF	Low-voltage detection flag
0	• LVISEL = 0: Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}), or when LVI operation is disabled • LVISEL = 1: Input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI}), or when LVI operation is disabled
1	• LVISEL = 0: Supply voltage (V_{DD}) < detection voltage (V_{LVI}) • LVISEL = 1: Input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI})

Notes 1. The reset value changes depending on the reset source and the setting of the option byte.
This register is not cleared (00H) by LVI reset.
It is set to “82H” when a reset signal other than LVI is applied if option byte LVIOFF = 0, and to “00H” if option byte LVIOFF = 1.

2. Bit 0 is read-only.

3. LVION, LVIMD, and LVISEL are cleared to 0 in the case of a reset other than an LVI reset. These are not cleared to 0 in the case of an LVI reset.

4. When LVION is set to 1, operation of the comparator in the LVI circuit is started. Use software to wait for the following periods of time, between when LVION is set to 1 and when the voltage is confirmed with LVIF.
- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))

The LVIF value for these periods may be set/cleared regardless of the voltage level, and can therefore not be used. Also, the LVIIF interrupt request flag may be set to 1 in these periods.

(**Cautions** are listed on the next page.)

Cautions 1. To stop LVI, be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

2. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

3. When LVI is used in interrupt mode (LVIMD = 0) and LVISEL is set to 0, an interrupt request signal (INTLVI) that disables LVI operation (clears LVION) when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}) (if LVISEL = 1, input voltage of external input pin (EXLVI) is less than or equal to the detection voltage (V_{EXLVI})) is generated and LVIIF may be set to 1.

(2) Low-voltage detection level select register (LVIS)

This register selects the low-voltage detection level.

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

Reset signal generation input sets this register to 0EH.

Figure 25-3. Format of Low-Voltage Detection Level Select Register (LVIS)

Address: FFFAAH After reset: 0EH[Note] R/W

Symbol	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	Detection level
0	1	0	1	V_{LVI0} (3.45 ±0.1 V)
0	1	1	0	V_{LVI1} (3.30 ±0.1 V)
0	1	1	1	V_{LVI2} (3.15 ±0.1 V)
1	0	0	0	V_{LVI3} (2.99 ±0.1 V)
1	0	0	1	V_{LVI4} (2.84 ±0.1 V)
1	0	1	0	V_{LVI5} (2.68 ±0.1 V)
1	0	1	1	V_{LVI6} (2.53 ±0.1 V)
1	1	0	0	V_{LVI7} (2.38 ±0.1 V)
1	1	0	1	V_{LVI8} (2.22 ±0.1 V)
1	1	1	0	V_{LVI9} (2.07 ±0.1 V)
1	1	1	1	V_{LVI10} (1.91 ±0.1 V)
Other above				Setting prohibited

Note The reset value changes depending on the reset source.
If the LVIS register is reset by LVI, it is not reset but holds the current value. The value of this register is reset to “0EH” if a reset other than by LVI is effected.

(**Cautions** are listed on the next page.)

Cautions 1. Be sure to clear bits 4 to 7 to "0".

2. Change the LVIS value with either of the following methods.

- **When changing the value after stopping LVI**
  - **<1> Stop LVI (LVION = 0).**
  - **<2> Change the LVIS register.**
  - **<3> Set to the mode used as an interrupt (LVIMD = 0).**
  - **<4> Mask LVI interrupts (LVIMK = 1).**
  - **<5> Enable LVI operation (LVION = 1).**
  - **<6> Before cancelling the LVI interrupt mask (LVIMK = 0), clear it with software because an LVIIF flag may be set when LVI operation is enabled.**
- **When changing the value after setting to the mode used as an interrupt (LVIMD = 0)**
  - **<1> Mask LVI interrupts (LVIMK = 1).**
  - **<2> Set to the mode used as an interrupt (LVIMD = 0).**
  - **<3> Change the LVIS register.**
  - **<4> Before cancelling the LVI interrupt mask (LVIMK = 0), clear it with software because an LVIIF flag may be set when the LVIS register is changed.**

3. When an input voltage from the external input pin (EXLVI) is detected, the detection voltage (V_{EXLVI}) is fixed. Therefore, setting of LVIS is not necessary.

(3) Port mode register 12 (PM12)

When using the P120/EXLVI/INTP0 pin for external low-voltage detection potential input, set PM120 to 1. At this time, the output latch of P120 may be 0 or 1.

PM12 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 25-4. Format of Port Mode Register 12 (PM12)

Address: FFF2CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM12	1	1	1	1	1	1	1	PM120

PM120	P120 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

25.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

(1) Used as reset (LVIMD = 1)

- If LVISEL = 0, compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), generates an internal reset signal when $V_{DD} < V_{LVI}$, and releases internal reset when $V_{DD} \geq V_{LVI}$.
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage (V_{EXLVI}), generates an internal reset signal when EXLVI < V_{EXLVI}, and releases internal reset when EXLVI ≥ V_{EXLVI}.

Remark The low-voltage detector (LVI) can be set to ON by an option byte by default. If it is set to ON to raise the power supply from the POC detection voltage (V_{POR} = 1.61 V (TYP.)) or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.2 V). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 V).

(2) Used as interrupt (LVIMD = 0)

- If LVISEL = 0, compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}). When V_{DD} drops lower than V_{LVI} ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$), generates an interrupt signal (INTLVI).
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage (V_{EXLVI} = 1.21 V ±0.1 V). When EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}), generates an interrupt signal (INTLVI).

While the low-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 low-voltage detection flag (LVIF: bit 0 of LVIM).

Remark LVIMD: Bit 1 of low-voltage detection register (LVIM)
LVISEL: Bit 2 of LVIM

25.4.1 When used as reset

(1) When detecting level of supply voltage (V_{DD})

(a) When LVI default start function stopped is set (LVIOFF = 1)

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD})) (default value).
  - <3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).
  - <4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <5> Use software to wait for the following periods of time (Total 210 μs).
    - Operation stabilization time (10 μs (MAX.))
    - Minimum pulse width (200 μs (MIN.))
  - <6> Wait until it is checked that (supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})) by bit 0 (LVIF) of LVIM.
  - <7> Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected).

  Figure 25-5 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

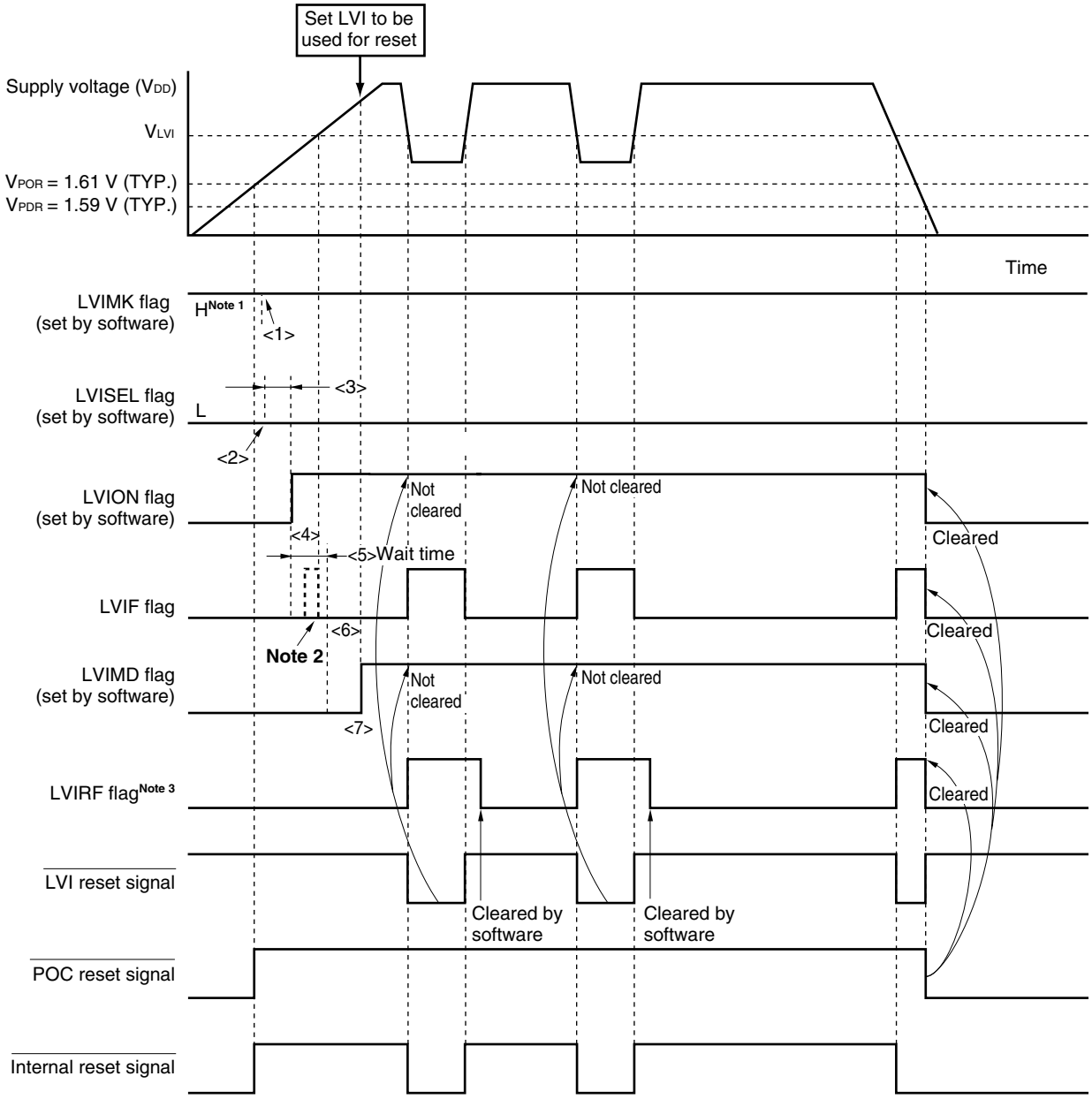
  Cautions 1. Be sure to execute <1>. When LVIMK = 0, an interrupt may occur immediately after the processing in <4>.

  2. If supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}) when LVIMD is set to 1, an internal reset signal is not generated.

- When stopping operation

  Be sure to clear (0) LVIMD and then LVION by using a 1-bit memory manipulation instruction.

Figure 25-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The LVIIF flag of the interrupt request flag registers and the LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 23 RESET FUNCTION**.

Remarks 1. <1> to <7> in Figure 25-5 above correspond to <1> to <7> in the description of "When starting operation" in **25.4.1 (1) (a) When LVI default start function stopped is set (LVIOFF = 1).**

2. V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(b) When LVI default start function enabled is set (LVIOFF = 0)

- When starting operation

  Start in the following initial setting state.

  - Set bit 7 (LVION) of LVIM to 1 (enables LVI operation)
  - Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD}))
  - Set the low-voltage detection level selection register (LVIS) to 0EH (default value: V_{LVI} = 2.07 V ±0.1 V).
  - Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected)
  - Set bit 0 (LVIF) of LVIM to 0 (“Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})”)

  Figure 25-6 shows the timing of the internal reset signal generated by the low-voltage detector.

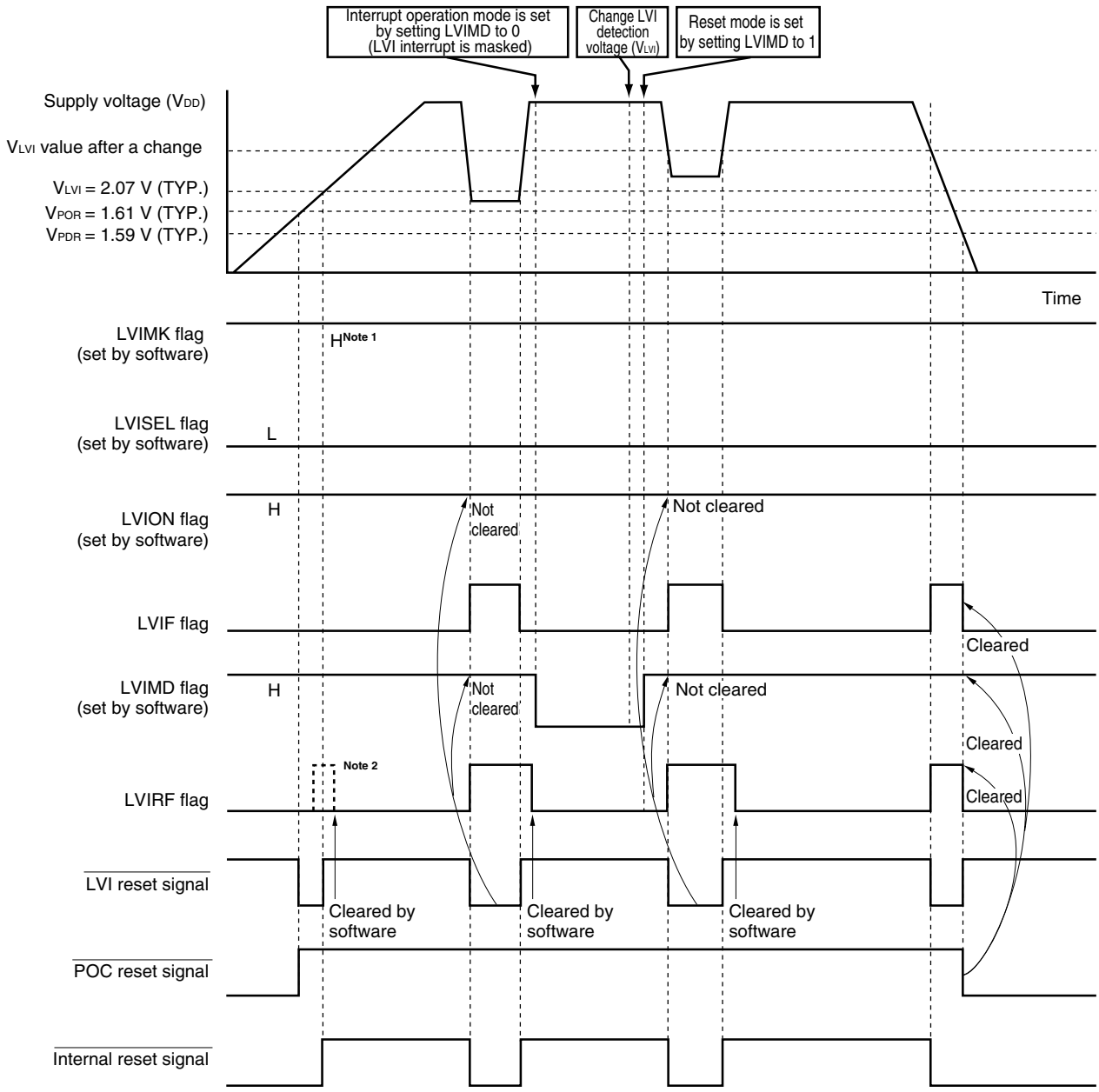
- When stopping operation

  Be sure to clear (0) LVIMD and then LVION by using a 1-bit memory manipulation instruction.

Caution Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- **Does not perform low-voltage detection during LVION = 0.**
- **If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution. This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.**

Figure 25-6. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 0)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. LVIRF is bit 0 of the reset control flag register (RESF).
When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, the LVIRF flag may become 1 from the beginning due to the power-on waveform.
For details of RESF, see **CHAPTER 23 RESET FUNCTION**.

Remark V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(2) When detecting level of input voltage from external input pin (EXLVI)

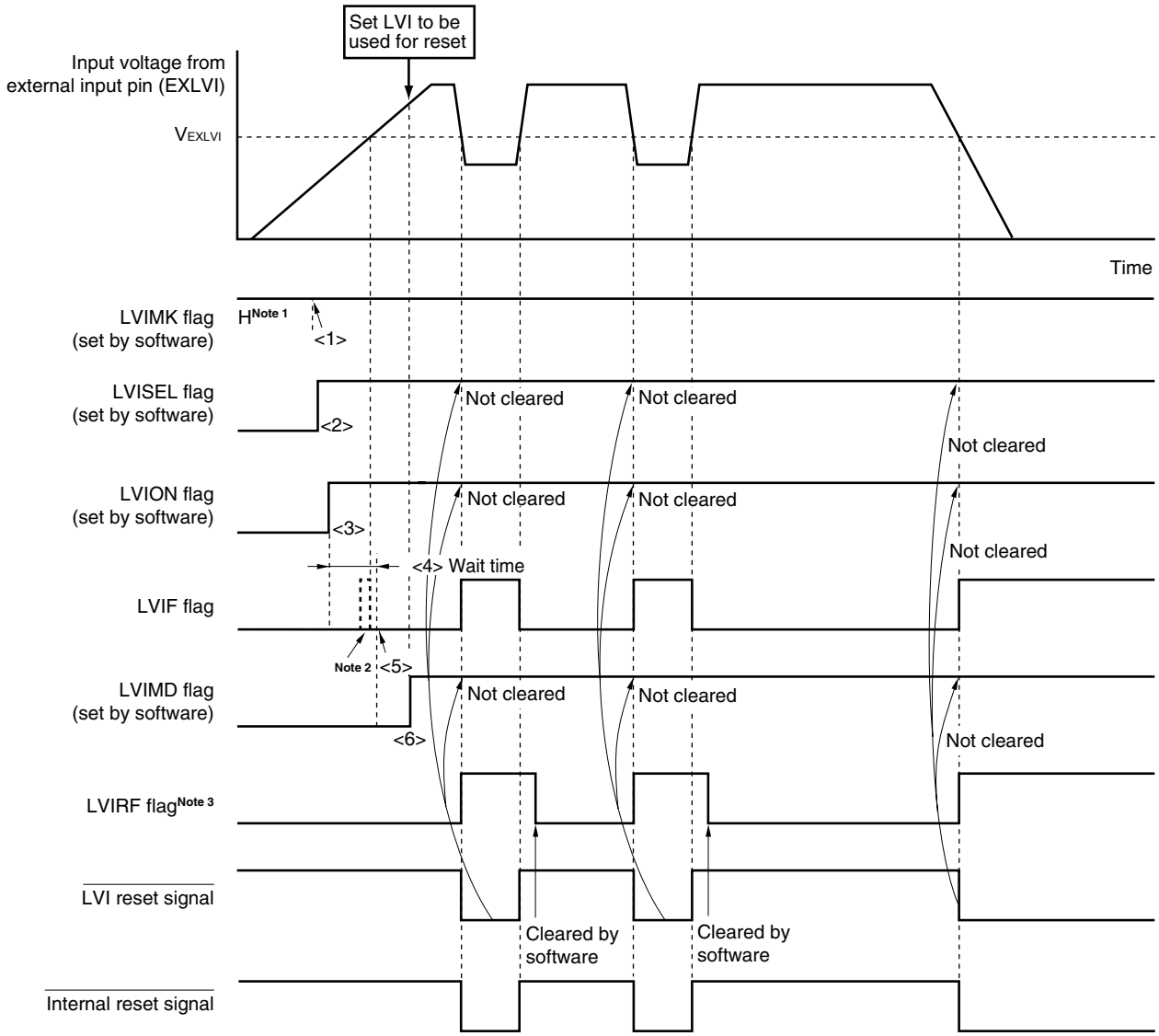
- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
  - <3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <4> Use software to wait for the following periods of time (Total 210 μs).
    - Operation stabilization time (10 μs (MAX.))
    - Minimum pulse width (200 μs (MIN.))
  - <5> Wait until it is checked that (input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.))) by bit 0 (LVIF) of LVIM.
  - <6> Set bit 1 (LVIMD) of LVIM to 1 (generates reset signal when the level is detected).

  Figure 25-7 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <6> above.

  Cautions 1. Be sure to execute <1>. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.
  2. If input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.
  3. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation
  Be sure to clear (0) LVIMD and then LVION by using a 1-bit memory manipulation instruction.

Figure 25-7. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The LVIIF flag of the interrupt request flag registers and the LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 23 RESET FUNCTION**.

Remark <1> to <6> in Figure 25-7 above correspond to <1> to <6> in the description of "When starting operation" in **25.4.1 (2) When detecting level of input voltage from external input pin (EXLVI)**.

25.4.2 When used as interrupt

(1) When detecting level of supply voltage (V_{DD})

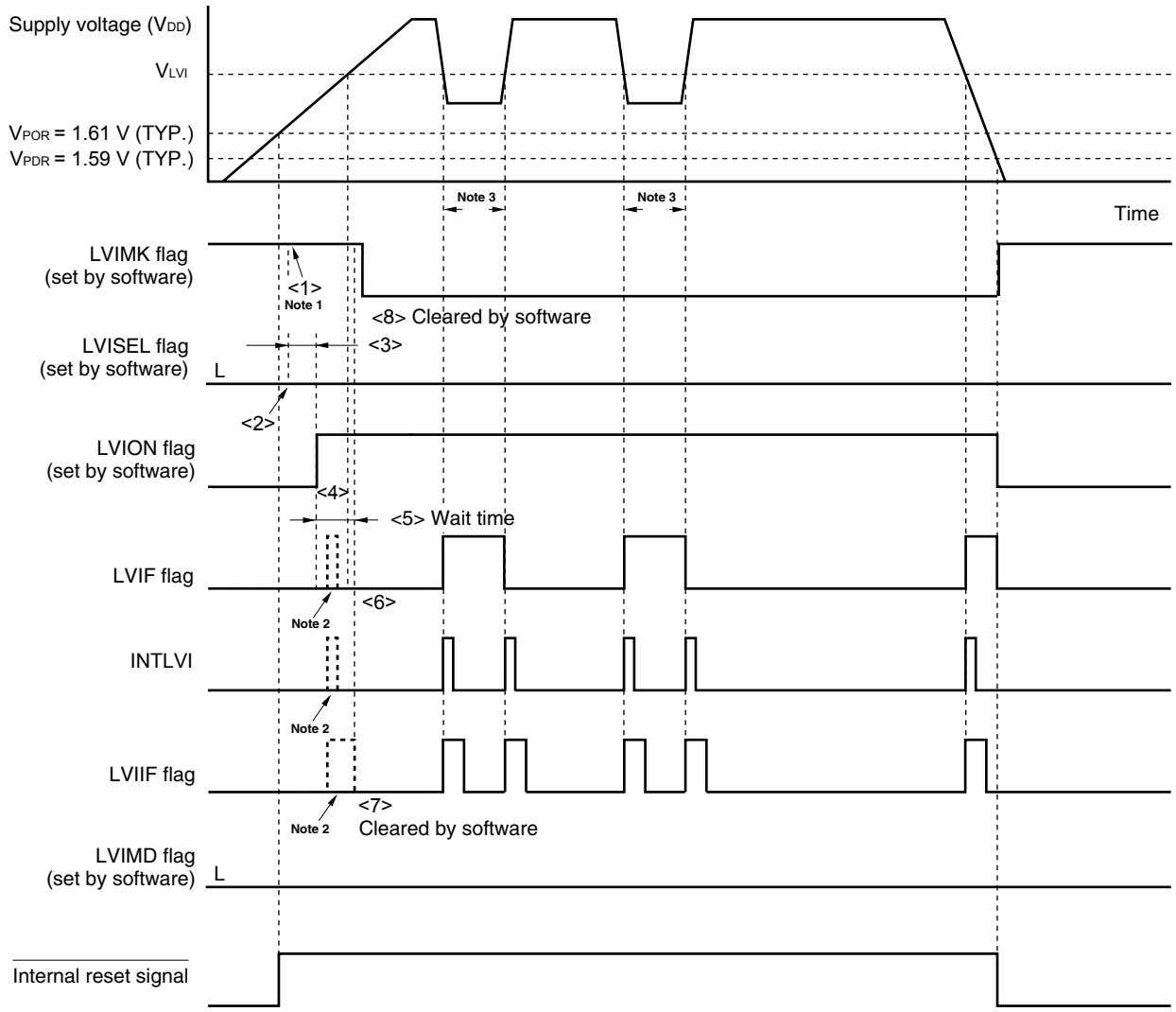
(a) When LVI default start function stopped is set (LVIOFF = 1)

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD})) (default value).
    Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).
  - <4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <5> Use software to wait for the following periods of time (Total 210 μs).
    - Operation stabilization time (10 μs (MAX.))
    - Minimum pulse width (200 μs (MIN.))
  - <6> Confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, at bit 0 (LVIF) of LVIM.
  - <7> Clear the interrupt request flag of LVI (LVIIF) to 0.
  - <8> Release the interrupt mask flag of LVI (LVIMK).
  - <9> Execute the EI instruction (when vector interrupts are used).

Figure 25-8 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <8> above.

- When stopping operation
  Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

Figure 25-8. Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVI operation is disabled when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

Remarks 1. <1> to <8> in Figure 25-8 above correspond to <1> to <8> in the description of "When starting operation" in **25.4.2 (1) (a) When LVI default start function stopped is set (LVIOFF = 1).**

2. V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(b) When LVI default start function enabled is set (LVIOFF = 0)

- When starting operation

<1> Start in the following initial setting state.

- Set bit 7 (LVION) of LVIM to 1 (enables LVI operation)
- Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD}))
- Set the low-voltage detection level selection register (LVIS) to 0EH (default value: V_{LVI} = 2.07 V ±0.1 V).
- Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected)
- Set bit 0 (LVIF) of LVIM to 0 (Detects falling edge "Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})")

<2> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).

<3> Release the interrupt mask flag of LVI (LVIMK).

<4> Execute the EI instruction (when vector interrupts are used).

Figure 25-9 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <3> above.

- When stopping operation

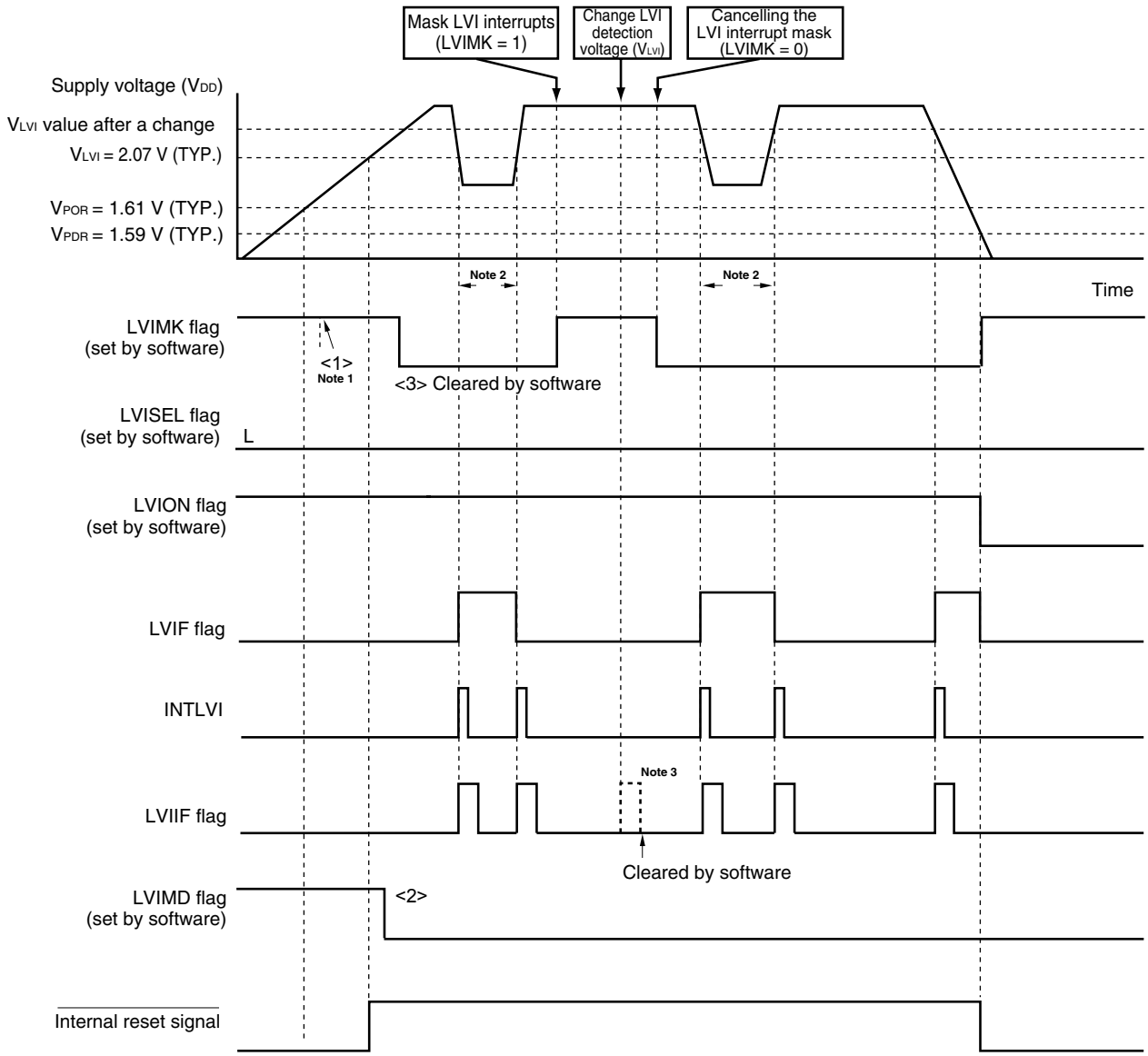
Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

Cautions 1. Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- **Does not perform low-voltage detection during LVION = 0.**
- **If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution.**
  This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.

2. When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, the LVIRF flag may become 1 from the beginning due to the power-on waveform.
For details of RESF, see CHAPTER 23 RESET FUNCTION.

Figure 25-9. Timing of Low-Voltage Detector Interrupt Signal Generation
(Bit: LVISEL = 0, Option Byte: LVIOFF = 0)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. If LVI operation is disabled when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

3. The LVIIF flag may be set when the LVI detection voltage is changed.

Remarks 1. <1> to <3> in Figure 25-9 above correspond to <1> to <3> in the description of "When starting operation" in **25.4.2 (1) (b) When LVI default start function enabled is set (LVIOFF = 0).**

2. V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(2) When detecting level of input voltage from external input pin (EXLVI)

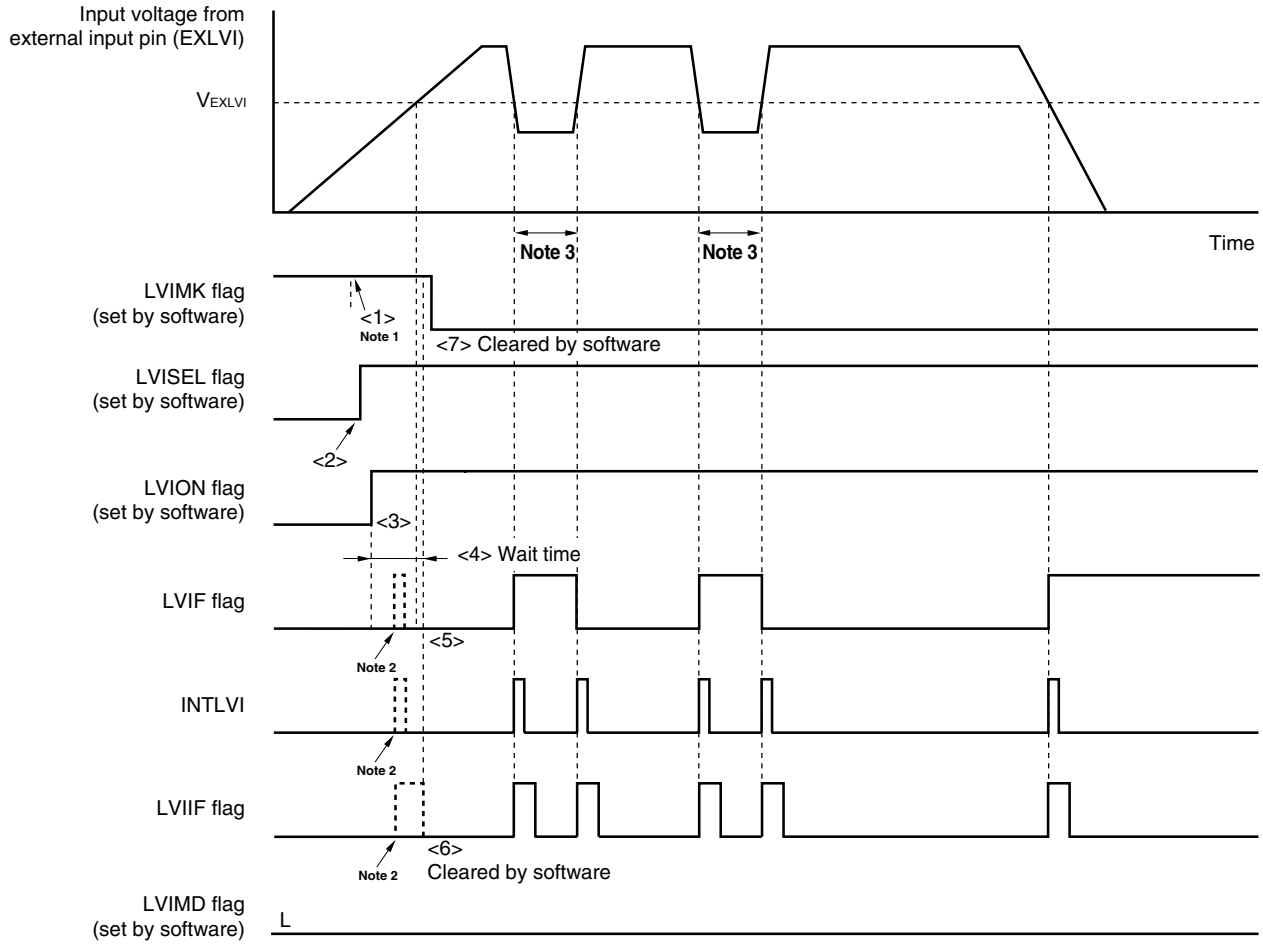
- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
    Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <4> Use software to wait for the following periods of time (Total 210 μs).
    - Operation stabilization time (10 μs (MAX.))
    - Minimum pulse width (200 μs (MIN.))
  - <5> Confirm that "input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.))" when detecting the falling edge of EXLVI, or "input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI} = 1.21 V (TYP.))" when detecting the rising edge of EXLVI, at bit 0 (LVIF) of LVIM.
  - <6> Clear the interrupt request flag of LVI (LVIIF) to 0.
  - <7> Release the interrupt mask flag of LVI (LVIMK).
  - <8> Execute the EI instruction (when vector interrupts are used).

  Figure 25-10 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

  Caution Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation
  Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

Figure 25-10. Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 1)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVI operation is disabled when the input voltage of external input pin (EXLVI) is less than or equal to the detection voltage (V_{EXLVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

Remark <1> to <7> in Figure 25-10 above correspond to <1> to <7> in the description of “When starting operation” in **25.4.2 (2) When detecting level of input voltage from external input pin (EXLVI)**.

25.5 Cautions for Low-Voltage Detector

(1) Measures method when supply voltage (V_{DD}) frequently fluctuates in the vicinity of the LVI detection voltage (V_{LVI})

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVI detection voltage (V_{LVI}), the operation is as follows depending on how the low-voltage detector is used.

Operation example 1: When used as reset

The system may be repeatedly reset and released from the reset status.

The time from reset release through microcontroller operation start can be set arbitrarily by the following action.

<Action>

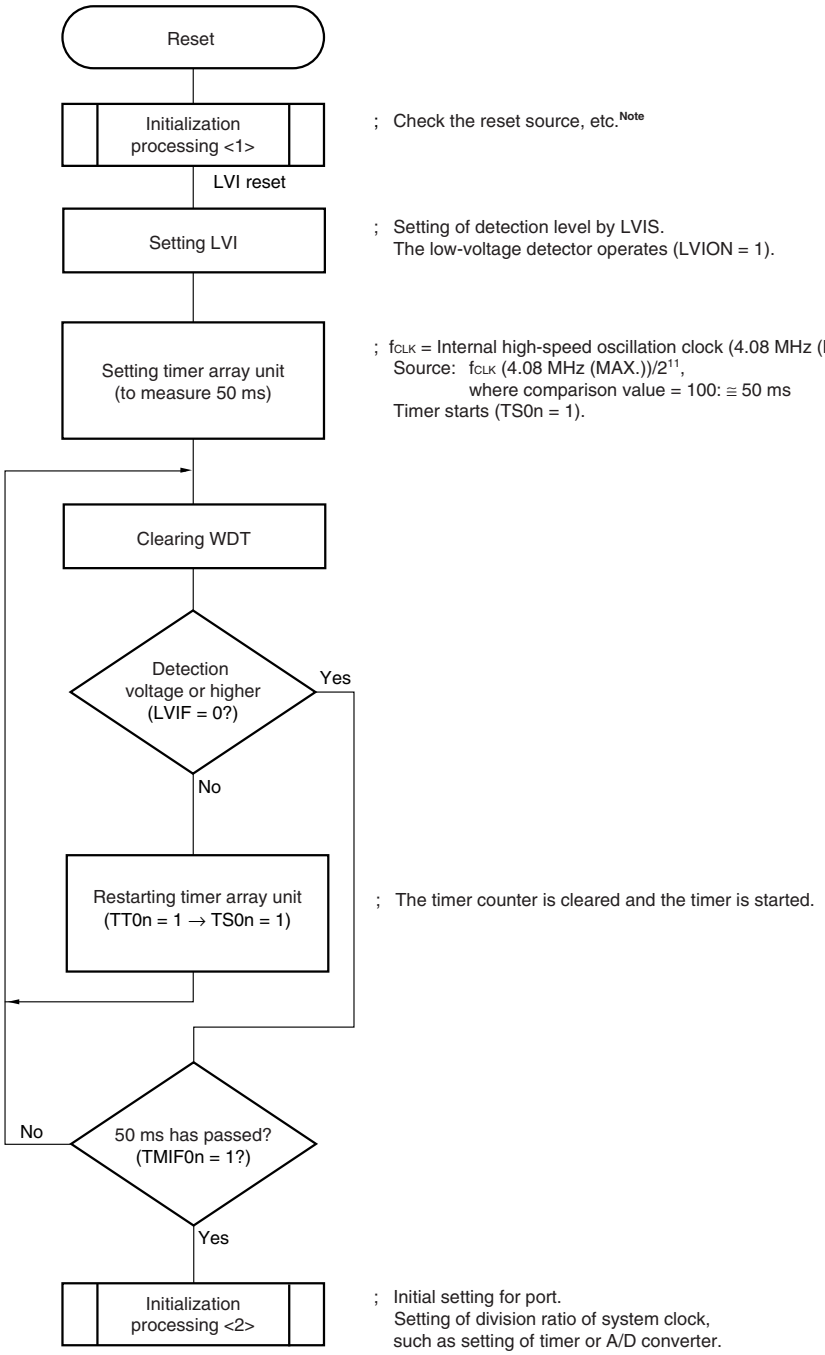
After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (see **Figure 25-11**).

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to “1”, the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

Figure 25-11. Example of Software Processing After Reset Release (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



Note A flowchart is shown on the next page.

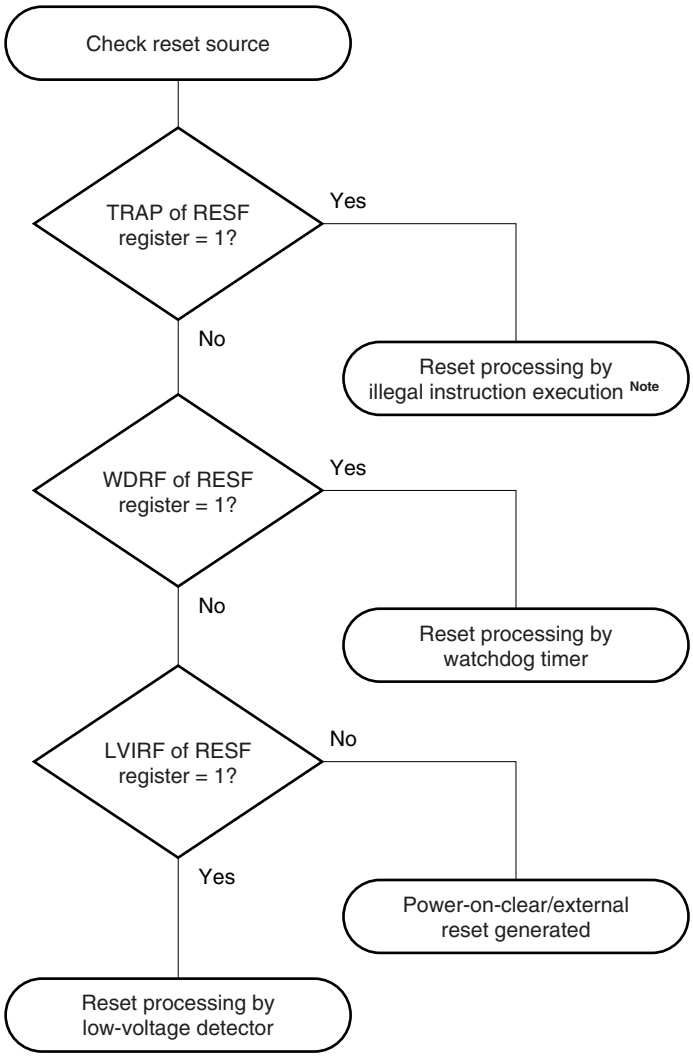
Remarks 1. If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

2. n = 0 to 7

Figure 25-11. Example of Software Processing After Reset Release (2/2)

• Checking reset source



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

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to “1”, the meanings of the above words change as follows.
- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

Operation example 2: When used as interrupt

Interrupt requests may be generated frequently.

Take the following action.

<Action>

Confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 1 (LVIIF) of interrupt request flag register 0L (IF0L) to 0.

For a system with a long supply voltage fluctuation period near the LVI detection voltage, take the above action after waiting for the supply voltage fluctuation time.

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

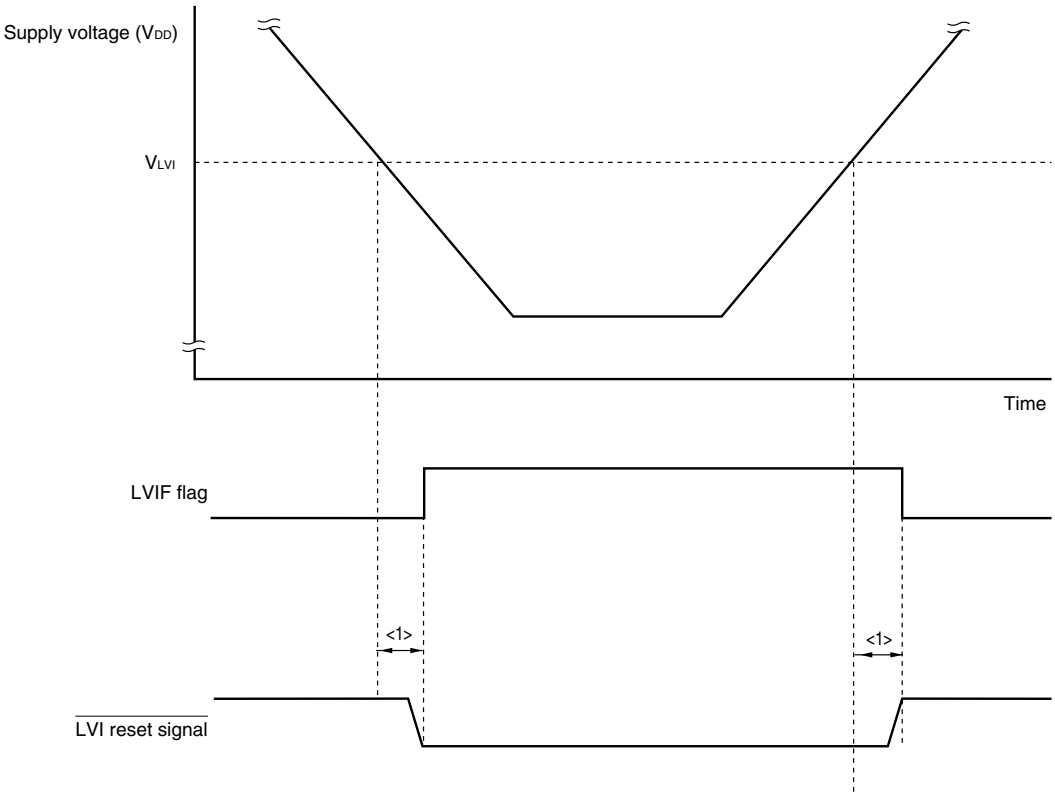
- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

(2) Delay from the time LVI reset source is generated until the time LVI reset has been generated or released

There is some delay from the time supply voltage (V_{DD}) < LVI detection voltage (V_{LVI}) until the time LVI reset has been generated.

In the same way, there is also some delay from the time LVI detection voltage (V_{LVI}) ≤ supply voltage (V_{DD}) until the time LVI reset has been released (see **Figure 25-12**).

Figure 25-12. Delay from the time LVI reset source is generated until the time LVI reset has been generated or released



<1>: Minimum pulse width (200 μs (MIN.))

CHAPTER 26 REGULATOR

26.1 Regulator Overview

All 78K0R/Lx3-M microcontroller products contain a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF).

Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

The regulator output voltage is normally 2.4 V (TYP.), and in the low-power consumption mode, 1.8 V (TYP.).

26.2 Registers Controlling Regulator

(1) Regulator mode control register (RMC)

This register sets the output voltage of the regulator.

RMC is set with an 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 26-1. Format of Regulator Mode Control Register (RMC)

Address: F00F4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMC								

RMC[7:0]	Control of output voltage of regulator
5AH	Fixed to low-power consumption mode (1.8 V)
00H	Switches normal power mode (2.4 V) and low-power consumption mode (1.8 V) according to the condition (refer to **Table 26-1**)
Other than above	Setting prohibited

Cautions 1. The RMC register can be rewritten only in the low-power consumption mode (refer to Table 26-1). In other words, rewrite this register during CPU operation with the subsystem clock (f_{XT}) while the high-speed system clock (f_{MX}), the high-speed internal oscillation clock, and the 20 MHz internal high-speed oscillation clock (f_{IH20}) are both stopped.

2. When using the setting fixed to the low consumption current mode, the RMC register can be used in the following cases.

<When the high-speed internal oscillation clock (f_{IH} = 8 MHz (TYP.) or f_{IH} = 1 MHz (TYP.)) is selected as the CPU clock>

f_{CLK} ≤ 1 MHz and external oscillator (X1 clock (f_X), external main system clock (f_{EX})) stop.

<When the X1 clock (f_X) or external main system clock (f_{EX}) is selected as the CPU clock>

f_{CLK} ≤ 1 MHz, f_X/f_{EX} ≤ 5 MHz and the internal high-speed oscillator stop.

<When the subsystem clock (f_{SUB}) is selected as the CPU clock>

Both the internal high-speed oscillator and external oscillator (f_X/f_{EX} ≤ 5 MHz) stop or either one stops.

Cautions 3. In low-power consumption mode, use the regulator with f_{CLK} fixed to 1 MHz when executing self programming.

4. A wait is required to change the operation speed mode control register (OSMC) after changing the RMC register. Wait for 2 ms by software when setting to low-power consumption mode and 10 μs when setting to normal power mode, as described in the procedure shown below.

• When setting to low-power consumption mode

<1> Select a frequency of 1 MHz for f_{CLK}.

<2> Set RMC to 5AH (set the regulator to low-power consumption mode).

<3> Wait for 2 ms.

<4> Set FLPC and FSEL of OSMC to 1 and 0, respectively.

• When setting to normal power mode

<1> Set RMC to 00H (set the regulator to normal power mode).

<2> Wait for 10 μs.

<3> Change FLPC and FSEL of OSMC.

<4> Change the f_{CLK} frequency.

Table 26-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
Low-power consumption mode	1.8 V	In STOP mode (except during OCD mode)
		When both the high-speed system clock (f_{MX}), the high-speed internal oscillation clock (f_{IH}), and the 20 MHz internal high-speed oscillation clock (f_{IH20}) are stopped during CPU operation with the subsystem clock (f_{SUB})
		When both the high-speed system clock (f_{MX}), the high-speed internal oscillation clock (f_{IH}), and the 20 MHz internal high-speed oscillation clock (f_{IH20}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{SUB}) has been set
Normal power mode	2.4 V	Other than above

CHAPTER 27 OPTION BYTE

27.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the 78K0R/Lx3-M microcontrollers 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.

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 Be sure to set FFH to 000C2H (000C2H/010C2H when the boot swap operation is used).

27.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)

(1) 000C0H/010C0H

- ○ Operation of watchdog timer
  - Operation is stopped or enabled in the HALT or STOP mode.
- ○ Setting of interval time of watchdog timer
- ○ Operation of watchdog timer
  - Operation is stopped or enabled.
- ○ Setting of window open period of watchdog timer
- ○ Setting of interval interrupt of watchdog timer
  - Used or not used

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 LVI upon reset release (upon power application)
  - LVI is ON or OFF by default upon reset release (reset by $\overline{\text{RESET}}$ pin excluding LVI, POC, WDT, or illegal instructions).
- ○ Setting of internal high-speed oscillator frequency
  - Select from 1 MHz, 8 MHz, or 20 MHz.

Caution Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

(3) 000C2H/010C2H

- ○ Be sure to set FFH, as these addresses are reserved areas.

Caution Set FFH 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[Note 1]

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINIT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period[Note 2]
0	0	Setting prohibited
0	1	50%
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 33 kHz (MAX.))
0	0	0	$2^7/f_{IL}$ (3.88 ms)
0	0	1	$2^8/f_{IL}$ (7.76 ms)
0	1	0	$2^9/f_{IL}$ (15.52 ms)
0	1	1	$2^{10}/f_{IL}$ (31.03 ms)
1	0	0	$2^{12}/f_{IL}$ (124.12 ms)
1	0	1	$2^{14}/f_{IL}$ (496.48 ms)
1	1	0	$2^{15}/f_{IL}$ (992.97 ms)
1	1	1	$2^{17}/f_{IL}$ (3971.88 ms)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode[Note 2]
1	Counter operation enabled in HALT/STOP mode

Notes 1. Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

2. The window open period is 100% when WDSTBYON = 0, regardless the value of WINDOW1 and WINDOW0.

Caution The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

Remark f_{IL}: Internal low-speed oscillation clock frequency

Figure 27-2. Format of User Option Byte (000C1H/010C1H)

Address: 000C1H/010C1H[Note 1]

7	6	5	4	3	2	1	0
1	1	1	1	1	FRQSEL2	FRQSEL1	LVIOFF

FRQSEL2	FRQSEL1	Internal high-speed oscillator frequency
0	1	8 MHz/20 MHz [Note 2]
1	0	1 MHz [Note 3]
1	1	8 MHz
Other than the above		Setting prohibited

LVIOFF	Setting of LVI on power application
0	LVI is ON by default (LVI default start function enabled) upon reset release (upon power application)
1	LVI is OFF by default (LVI default start function stopped) upon reset release (upon power application)

Notes 1. Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

2. When 8 MHz or 20 MHz has been selected, the 8 MHz internal high-speed oscillator automatically starts oscillating after reset release. To use the 20 MHz internal high-speed oscillator to operate the microcontroller, oscillation is started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1 with $V_{DD} \geq 2.7$ V. The circuit cannot be changed to a 1 MHz internal high-speed oscillator while the microcontroller operates.

3. When 1 MHz has been selected, the microcontroller operates on the 1 MHz internal high-speed oscillator after reset release. The circuit cannot be changed to an 8 MHz or 20 MHz internal high-speed oscillator while the microcontroller operates.

Cautions 1. Be sure to set bits 7 to 3 to "1".

2. Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- **Does not perform low-voltage detection during LVION = 0.**
- **If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution.**
  This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.

Figure 27-3. Format of Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H[Note]

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

Note Be sure to set FFH to 000C2H, as these addresses are reserved areas. Also set FFH to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

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[Note]

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Erases data of flash memory in case of failures in enabling on-chip debugging and authenticating on-chip debug security ID.
1	1	Does not erases data of flash memory in case of failures in enabling on-chip debugging and authenticating on-chip debug security ID.

Note Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

Caution Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value.
Be sure to set 000010B to bits 6 to 1.

Remark The value on bits 3 to 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.
However, be sure to set the default values (0, 1, and 0) to bits 3 to 1 at setting.

27.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the RA78K0R or PM+ linker option, in addition to describing to the source. When doing so, the contents set by using the linker option take precedence, even if descriptions exist in the source, as mentioned below.

See the RA78K0R Assembler Package User's Manual for how to set the linker option.

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 2^10/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
      DB          0FBH    ; Select 8 MHz or 20 MHz for internal high-speed oscillator
                          ; Stops LVI default start function
      DB          0FFH    ; Reserved area
      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 2^10/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
      DB          0FBH    ; Select 8 MHz or 20 MHz for internal high-speed oscillator
                          ; Stops LVI default start function
      DB          0FFH    ; Reserved area
      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 78K0R/Lx3-M microcontrollers incorporate the flash memory to which a program can be written, erased, and overwritten while mounted on the board.

28.1 Writing with Flash Memory Programmer

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 78K0R/Lx3-M microcontrollers have 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

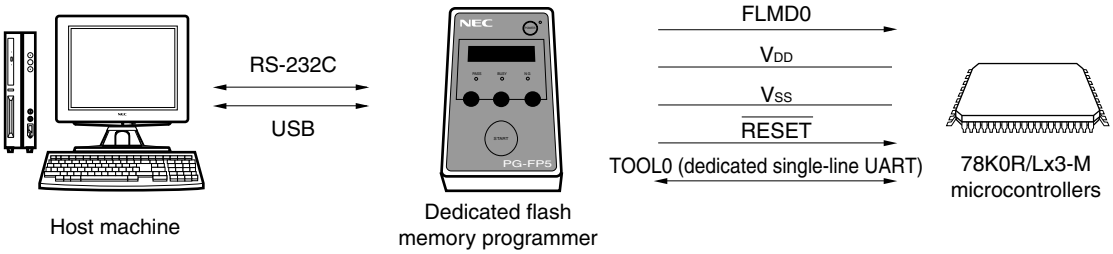
Data can be written to the flash memory with a dedicated program adapter (FA series) before the 78K0R/Lx3-M microcontrollers are mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

28.2 Programming Environment

The environment required for writing a program to the flash memory of the 78K0R/Lx3-M microcontrollers are illustrated below.

Figure 28-1. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

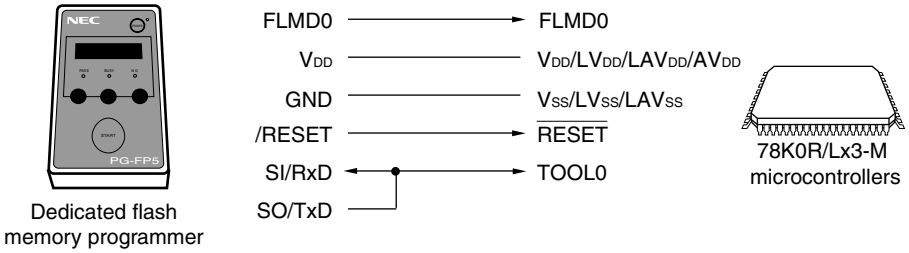
To interface between the dedicated flash memory programmer and the 78K0R/Lx3-M microcontrollers, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

28.3 Communication Mode

Communication between the dedicated flash memory programmer and the 78K0R/Lx3-M microcontrollers is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the 78K0R/Lx3-M microcontrollers.

Transfer rate: 115,200 bps to 1,000,000 bps

Figure 28-2. Communication with Dedicated Flash Memory Programmer



When using the FlashPro5 as the dedicated flash memory programmer, the FlashPro5 generates the following signals for the 78K0R/Lx3-M microcontrollers. For details, refer to the user's manual for the FlashPro5.

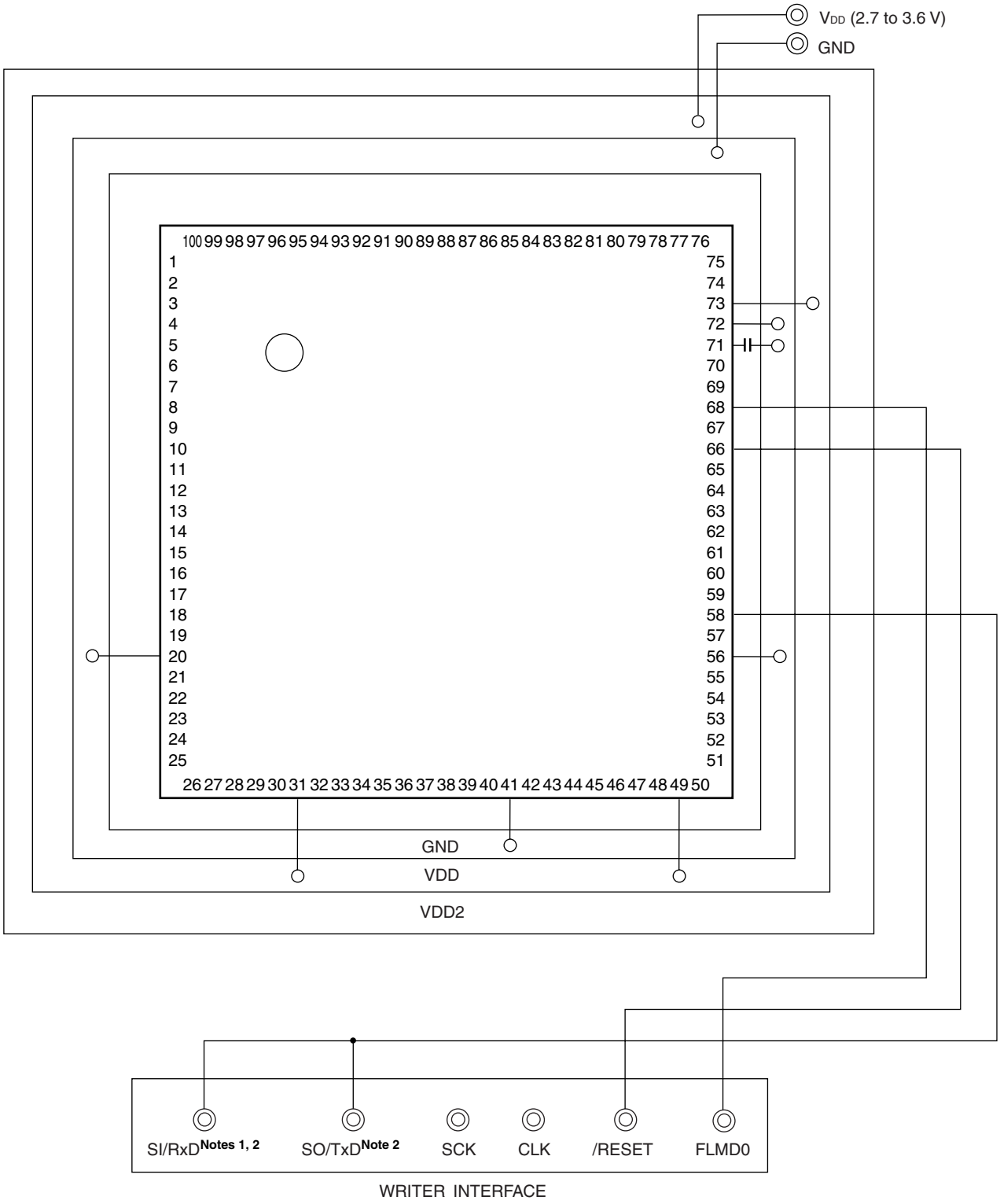
Table 28-1. Pin Connection

FlashPro5			78K0R/Lx3-M microcontrollers	Connection
Signal Name	I/O	Pin Function	Pin Name	
FLMD0	Output	Mode signal	FLMD0	◎
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, LV_{DD}, LAV_{DD}, AV_{DD}	◎
GND	–	Ground	V_{SS}, LV_{SS}, LAV_{SS}	◎
CLK	Output	Clock output	–	×
/RESET	Output	Reset signal	RESET	◎
SI/RxD	Input	Receive signal	TOOL0	◎
SO/TxD	Output	Transmit signal		
SCK	Output	Transfer clock	–	×

Remark ◎: Be sure to connect the pin.
×: The pin does not have to be connected.

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

Figure 28-3. Example of Wiring Adapter for Flash Memory Writing



Notes 1. This pin is not required to be connected when using PG-FP5 or FL-PR5.

2. Connect SI/RxD or SO/TxD when using QB-MINI2.

28.4 Connection of Pins on Board

To write the flash memory on-board, 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.

28.4.1 FLMD0 pin

(1) In flash memory programming mode

Directly connect this pin to a flash memory programmer when data is written by the flash memory programmer. This supplies a writing voltage of the V_{DD} level to the FLMD0 pin.

The FLMD0 pin does not have to be pulled down externally because it is internally pulled down by reset. To pull it down externally, use a resistor of 1 kΩ to 200 kΩ.

(2) In normal operation mode

It is recommended to leave this pin open during normal operation.

The FLMD0 pin must always be kept at the V_{SS} level before reset release but does not have to be pulled down externally because it is internally pulled down by reset. However, pulling it down must be kept selected (i.e., FLMDPUP = "0", default value) by using bit 7 (FLMDPUP) of the background event control register (BECTL) (see **28.5 (1) Back ground event control register**). To pull it down externally, use a resistor of 200 kΩ or smaller.

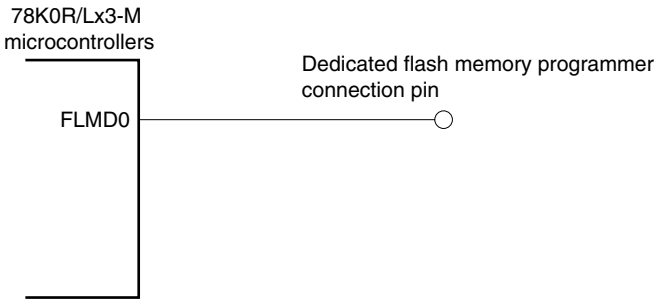
Self programming and the rewriting of flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V_{SS} pin.

(3) In self programming mode

It is recommended to leave this pin open when using the self programming function. To pull it down externally, use a resistor of 100 kΩ to 200 kΩ.

In the self programming mode, the setting is switched to pull up in the self programming library.

Figure 28-4. FLMD0 Pin Connection Example



28.4.2 TOOL0 pin

In the flash memory programming mode, connect this pin directly to the dedicated flash memory programmer or pull it up by connecting it to V_{DD} via an external resistor.

When on-chip debugging is enabled in the normal operation mode, pull this pin up by connecting it to V_{DD} via an external resistor, and be sure to keep inputting the V_{DD} level to the TOOL0 pin before reset is released (pulling down this pin is prohibited).

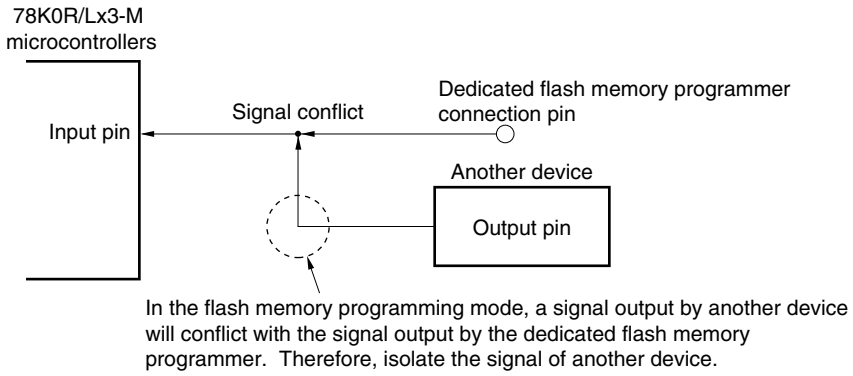
Remark The SAU and IICA pins are not used for communication between the 78K0R/Lx3-M microcontrollers and dedicated flash memory programmer, because single-line UART is used.

28.4.3 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer is 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.

Figure 28-5. Signal Conflict ($\overline{\text{RESET}}$ Pin)



28.4.4 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 V_{DD} or V_{SS} via a resistor.

28.4.5 REGC pin

Connect the REGC pin to GND via a capacitor (0.47 to 1 μF) in the same manner as during normal operation.

Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

28.4.6 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 internal high-speed oscillation clock (f_{IH}) is used.

28.4.7 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 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.

Supply the same other power supplies (LV_{DD}, LV_{SS}, LAV_{DD}, LAV_{SS}, and AV_{DD}) as those in the normal operation mode.

28.5 Registers Controlling Flash Memory

(1) Background event control register (BECTL)

Even if the FLMD0 pin is not controlled externally, it can be controlled by software with the BECTL register to set the self-programming mode.

However, depending on the processing of the FLMD0 pin, it may not be possible to set the self-programming mode by software. When using BECTL, leaving the FLMD0 pin open is recommended. When pulling it down externally, use a resistor with a resistance of 100 kΩ or more. In addition, in the normal operation mode, use BECTL with the pull down selection. In the self-programming mode, the setting is switched to pull up in the self- programming library.

The BECTL register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 28-6. Format of Background Event Control Register (BECTL)

Address: FFFBEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BECTL	FLMDPUP	0	0	0	0	0	0	0

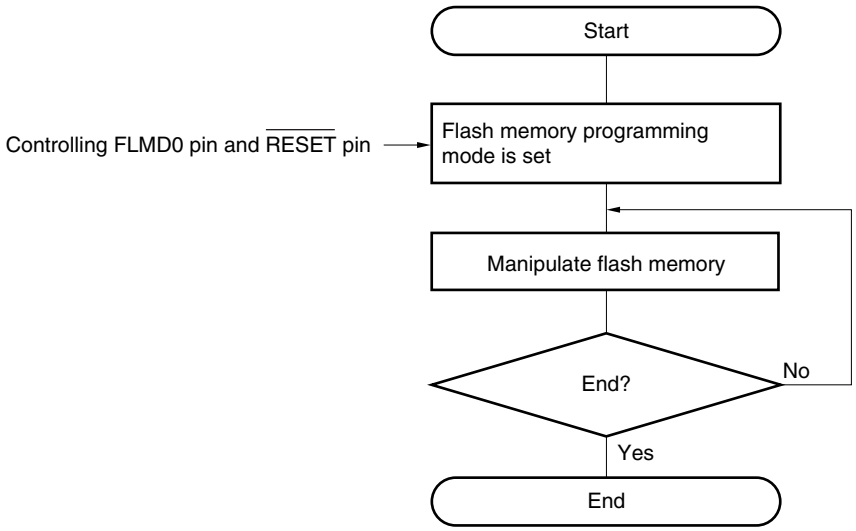
FLMDPUP	Software control of FLMD0 pin
0	Selects pull-down
1	Selects pull-up

28.6 Programming Method

28.6.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 28-7. Flash Memory Manipulation Procedure



28.6.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the 78K0R/Lx3-M microcontrollers in the flash memory programming mode. To set the mode, set the FLMD0 pin and TOOL0 pin to V_{DD} and clear the reset signal.

Change the mode by using a jumper when writing the flash memory on-board.

Figure 28-8. Flash Memory Programming Mode

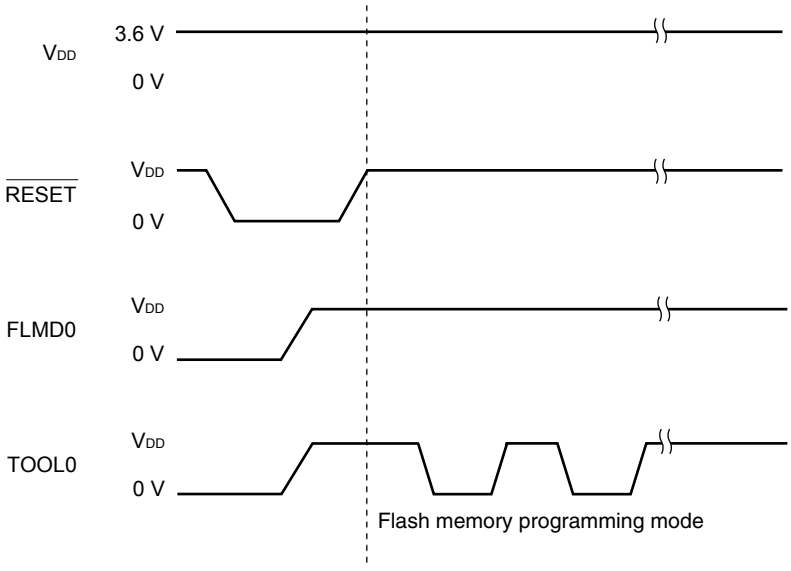


Table 28-2. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0	Normal operation mode
V_{DD}	Flash memory programming mode

28.6.3 Selecting communication mode

Communication mode of the 78K0R/Lx3-M microcontrollers as follows.

Table 28-3. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed	Frequency	Multiply Rate	
1-line mode (dedicated single-line UART)	UART-ch0	1 Mbps[Note 2]	–	–	TOOL0

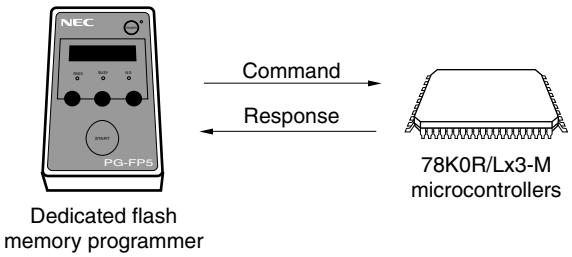
Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

28.6.4 Communication commands

The 78K0R/Lx3-M microcontrollers communicate with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0R/Lx3-M microcontrollers are called commands, and the signals sent from the 78K0R/Lx3-M microcontrollers to the dedicated flash memory programmer are called response.

Figure 28-9. Communication Commands



The flash memory control commands of the 78K0R/Lx3-M microcontrollers are listed in the table below. All these commands are issued from the programmer, and the 78K0R/Lx3-M microcontrollers perform processing corresponding to the respective commands.

Table 28-4. 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	Chip Erase	Erases the entire flash memory.
	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Write	Programming	Writes data to a specified area in the flash memory.
Getting information	Silicon Signature	Gets 78K0R/Lx3-M microcontrollers information (such as the part number and flash memory configuration).
	Version Get	Gets the 78K0R/Lx3-M microcontrollers firmware version.
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

The 78K0R/Lx3-M microcontrollers return a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0R/Lx3-M microcontrollers are listed below.

Table 28-5. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

28.7 Security Settings

The 78K0R/Lx3-M microcontrollers support 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. The security setting is valid when the programming mode is set next.

- Disabling batch erase (chip erase)
  Execution of the block erase and batch erase (chip erase) commands for entire blocks in the flash memory is prohibited by this setting during on-board/off-board programming. Once execution of the batch erase (chip erase) command is prohibited, all of the prohibition settings (including prohibition of batch erase (chip erase)) can no longer be cancelled.

  Caution After the security setting for the batch erase is set, erasure cannot be performed for the device. In addition, even if a write command is executed, data different from that which has already been written to the flash memory cannot be written, because the erase command is disabled.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during on-board/off-board programming. However, blocks can be erased by means of self programming.

- Disabling write
  Execution of the write and block erase commands for entire blocks in the flash memory is prohibited during on-board/off-board programming. However, blocks can be written by means of self programming.

- Disabling rewriting boot cluster 0
  Execution of the batch erase (chip erase) command, block erase command, and write command on boot cluster 0 (00000H to 00FFFH) in the flash memory is prohibited by this setting.

  Caution If a security setting that rewrites boot cluster 0 has been applied, boot cluster 0 of that device will not be rewritten, and the entire flash memory of the device will not be erased in batch.

The batch erase (chip erase), 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 on-board/off-board programming and self programming. Each security setting can be used in combination.

All the security settings are cleared by executing the batch erase (chip erase) command.

Table 28-6 shows the relationship between the erase and write commands when the 78K0R/Lx3-M microcontrollers security function is enabled.

Remark To prohibit writing and erasing during self-programming, use the flash sealed window function (see **28.8.2** for detail).

Table 28-6. Relationship Between Enabling Security Function and Command

(1) During on-board/off-board programming

Valid Security	Executed Command		
	Batch Erase (Chip Erase)	Block Erase	Write
Prohibition of batch erase (chip erase)	Cannot be erased in batch	Blocks cannot be erased.	Can be performed[Note].
Prohibition of block erase	Can be erased in batch.		Can be performed.
Prohibition of writing			Cannot be performed.
Prohibition of rewriting boot cluster 0	Cannot be erased in batch	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 batch erase (chip 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 batch erase (chip erase)	Blocks can be erased.	Can be performed.
Prohibition of block erase		
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 sealed window function (see **28.8.2** for detail).

Table 28-7. Setting Security in Each Programming Mode

(1) On-board/off-board programming

Security	Security Setting	How to Disable Security Setting
Prohibition of batch erase (chip erase)	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command
Prohibition of writing		
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

(2) Self programming

Security	Security Setting	How to Disable Security Setting
Prohibition of batch erase (chip erase)	Set by using information library.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command during on-board/off-board programming (cannot be disabled during self programming)
Prohibition of writing		
Prohibition of rewriting boot cluster 0		

28.8 Flash Memory Programming by Self-Programming

The 78K0R/Lx3-M microcontrollers support a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using the 78K0R/Lx3-M microcontrollers self-programming library, it can be used to upgrade the program in the field.

If an interrupt occurs during self-programming, self-programming can be temporarily stopped and interrupt servicing can be executed. If an unmasked interrupt request is generated in the EI state, the request branches directly from the self-programming library to the interrupt routine. After the self-programming mode is later restored, self-programming can be resumed. However, the interrupt response time is different from that of the normal operation mode.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.

2. In the self-programming mode, call the self-programming start library (FlashStart).

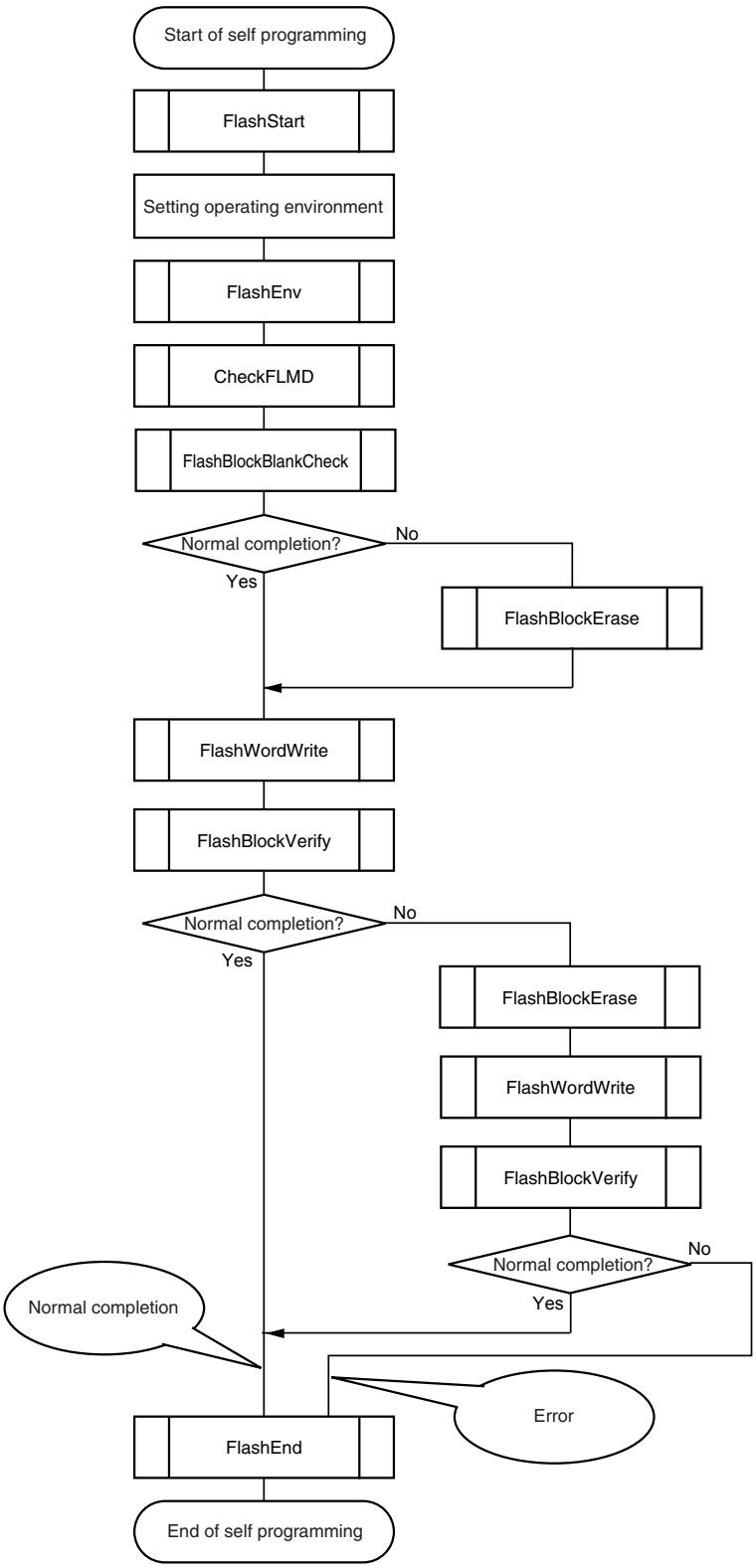
3. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the self-programming library.

4. In low-power-consumption mode, use the regulator with f_{CLK} fixed to 1 MHz when executing self programming. For details of the low-power-consumption mode, see CHAPTER 26 REGULATOR.

5. Disable DMA operation (DENn = 0) during the execution of self programming library functions.

The following figure illustrates a flow of rewriting the flash memory by using a self programming library.

Figure 28-10. Flow of Self Programming (Rewriting Flash Memory)



28.8.1 Boot swap function

If rewriting the boot area failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.

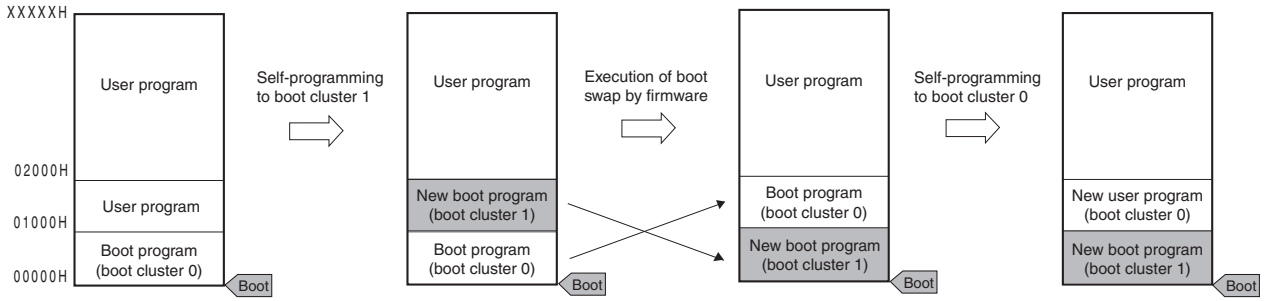
The boot swap function is used to avoid this problem.

Before erasing boot cluster 0[Note], which is a boot program 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 78K0R/Lx3-M microcontrollers, so that boot cluster 1 is used as a boot area. After that, erase or write the original boot program area, boot cluster 0.

As a result, even if a power failure occurs while the boot programming 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 28-11. Boot Swap Function

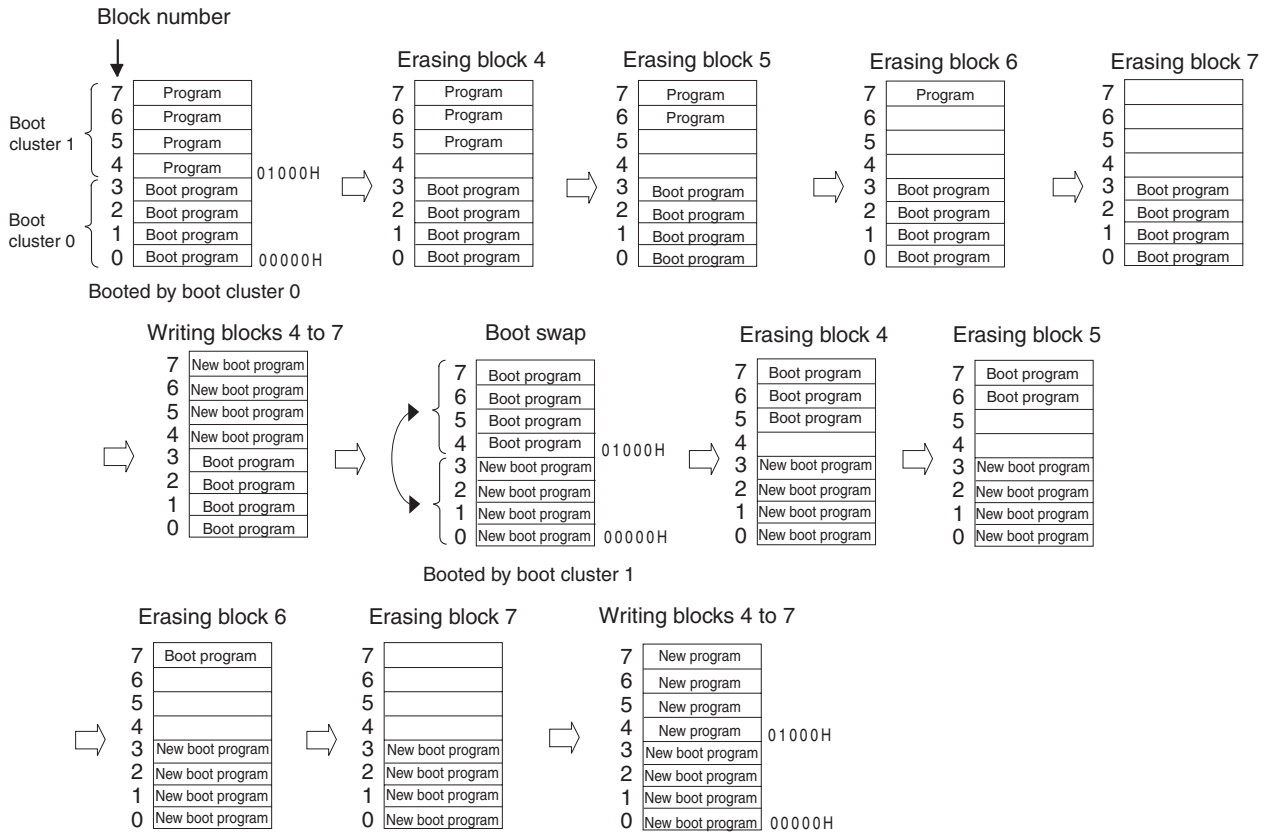


In an example of above figure, it is as follows.

Boot cluster 0: Boot program area before boot swap

Boot cluster 1: Boot program area after boot swap

Figure 28-12. Example of Executing Boot Swapping



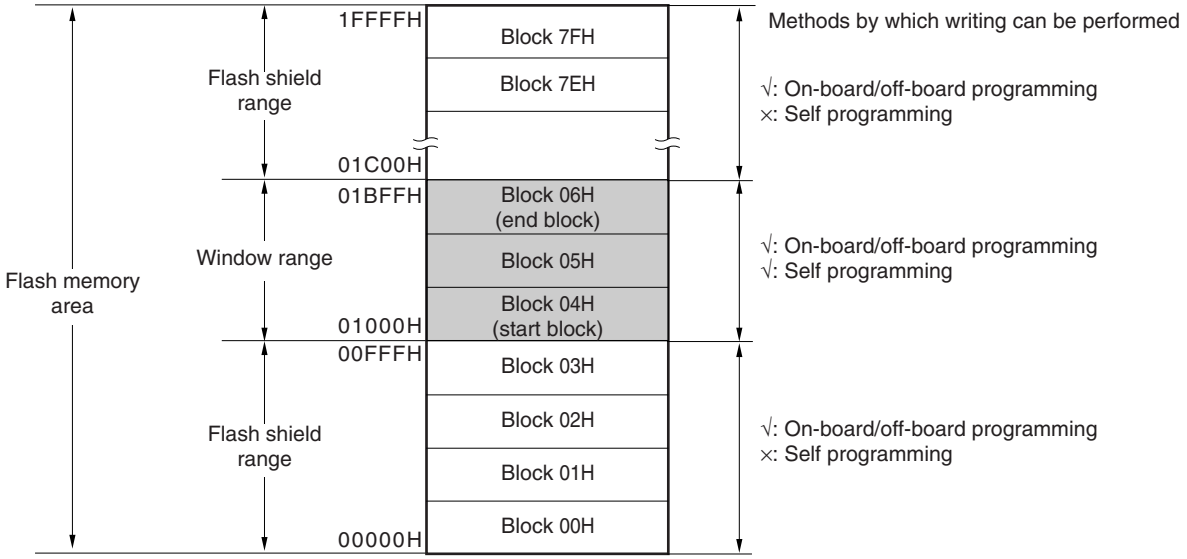
28.8.2 Flash shield window function

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 on-board/off-board programming and self programming.

Writing to and erasing areas outside the window range are disabled during self programming. During on-board/off-board programming, however, areas outside the range specified as a window can be written and erased.

Figure 28-13. Flash Shield Window Setting Example
(Target Devices: μPD78F8070, Start Block: 04H, End Block: 06H)



Caution 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.

Table 28-8. 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 set information library.	Block erasing is enabled only within the window range.	Writing is enabled only within the range of window range.
On-board/Off-board 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 **28.7 Security Settings** to prohibit writing/erasing during on-board/off-board programming.

28.9 Creating ROM Code to Place Order for Previously Written Product

Before placing an order with Renesas Electronics for a previously written product, the ROM code for the order must be created.

To create the ROM code, use the Hex Consolidation Utility (hereafter abbreviated to HCU) on the finished programs (hex files) and optional data (such as security settings for flash memory programs).

The HCU is a software tool that includes functions required for creating ROM code.

The HCU can be downloaded at the Renesas Electronics website.

(1) Website

https://secure-resource.renesas.com/micro/tool_reg/OdsListTop.do?lang=en

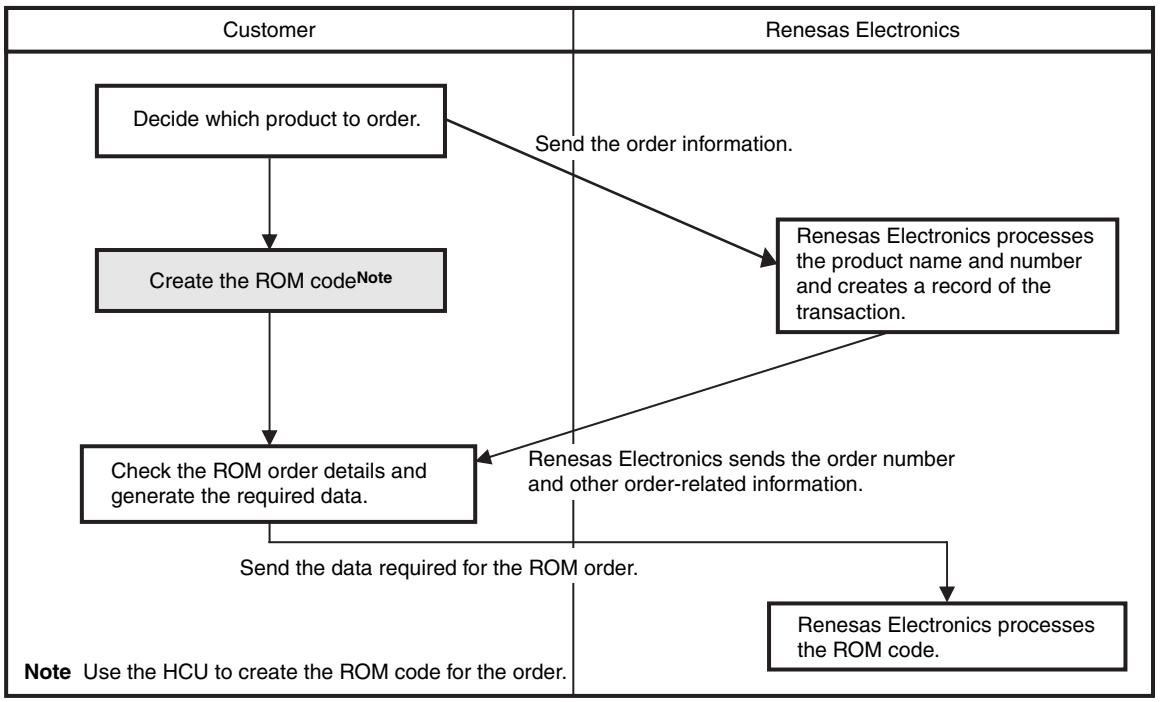
(2) Downloading the HCU

To download the HCU, click Software for previously written flash products and then HCU_GUI.

Remark For details about how to install and use the HCU, see the materials (the user's manual) that comes with the HCU at the above website.

28.9.1 Procedure for using ROM code to place an order

Use the HCU to create the ROM code by following the procedure below, and then place your order with Renesas Electronics. For details, see the ROM Code Ordering Method Information (C10302J).



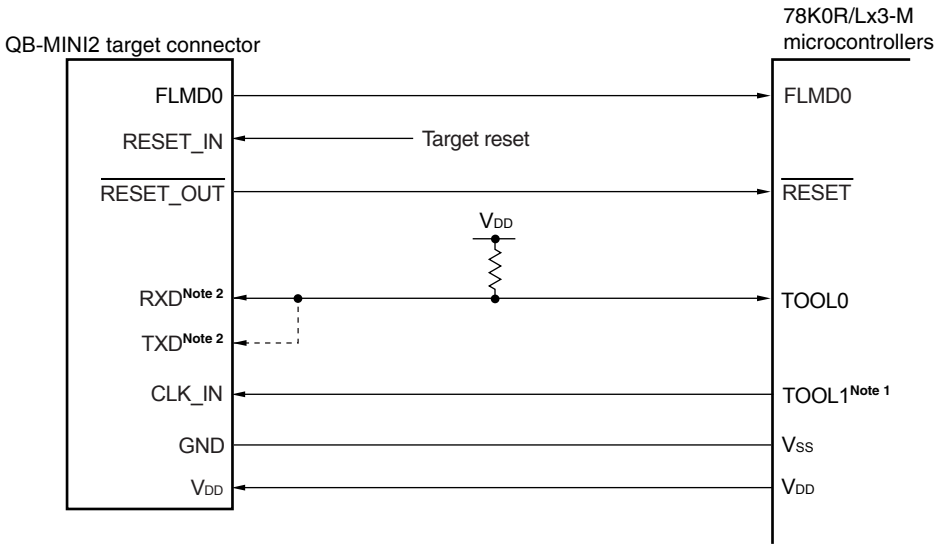
CHAPTER 29 ON-CHIP DEBUG FUNCTION

29.1 Connecting QB-MINI2 to 78K0R/Lx3-M microcontrollers

The 78K0R/Lx3-M microcontrollers use the V_{DD}, FLMD0, $\overline{RESET}$, TOOL0, TOOL1[Note 1], and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2).

Caution The 78K0R/Lx3-M 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.

Figure 29-1. Connection Example of QB-MINI2 and 78K0R/Lx3-M Microcontrollers



Notes 1. Connection is not required for communication in 1-line mode but required for communication in 2-line mode. At this time, perform necessary connections according to **Table 2-2 Connection of Unused Pins** since TOOL1 is an unused pin when QB-MINI2 is unconnected.

2. Connecting the dotted line is not necessary since RXD and TXD are shorted within QB-MIN2. When using the other flash memory programmer, RXD and TXD may not be shorted within the programmer. In this case, they must be shorted on the target system.

Remark The FLMD0 pin is recommended to be open for self-programming in on-chip debugging. To pull down externally, use a resistor of 100 kΩ or more.

1-line mode (single line UART) using the TOOL0 pin or 2-line mode using the TOOL0 and TOOL1 pins is used for serial communication For flash memory programming, 1-line mode is used. 1-line mode or 2-line mode is used for on-chip debugging. Table 29-1 lists the differences between 1-line mode and 2-line mode.

Table 29-1. Lists the Differences Between 1-line Mode and 2-line Mode.

Communication mode	Flash memory programming function	Debugging function
1-line mode	Available	• Pseudo real-time RAM monitor (RRM) function not supported. • DMM function (rewriting memory in RUN) not supported. • The debugger speed is two to four times slower than 2-line mode.
2-line mode	None	• Pseudo real-time RAM monitor (RRM) function supported • DMM function (rewriting memory in RUN) supported

Remark 2-line mode is not used for flash programming, however, even if TOOL1 pin is connected with CLK_IN of QB-MINI2, writing is performed normally with no problem.

29.2 On-Chip Debug Security ID

The 78K0R/Lx3-M microcontrollers have 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.

For details on the on-chip debug security ID, refer to the **QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual (U18371E)**.

Table 29-2. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
010C4H to 010CDH	

29.3 Securing of User Resources

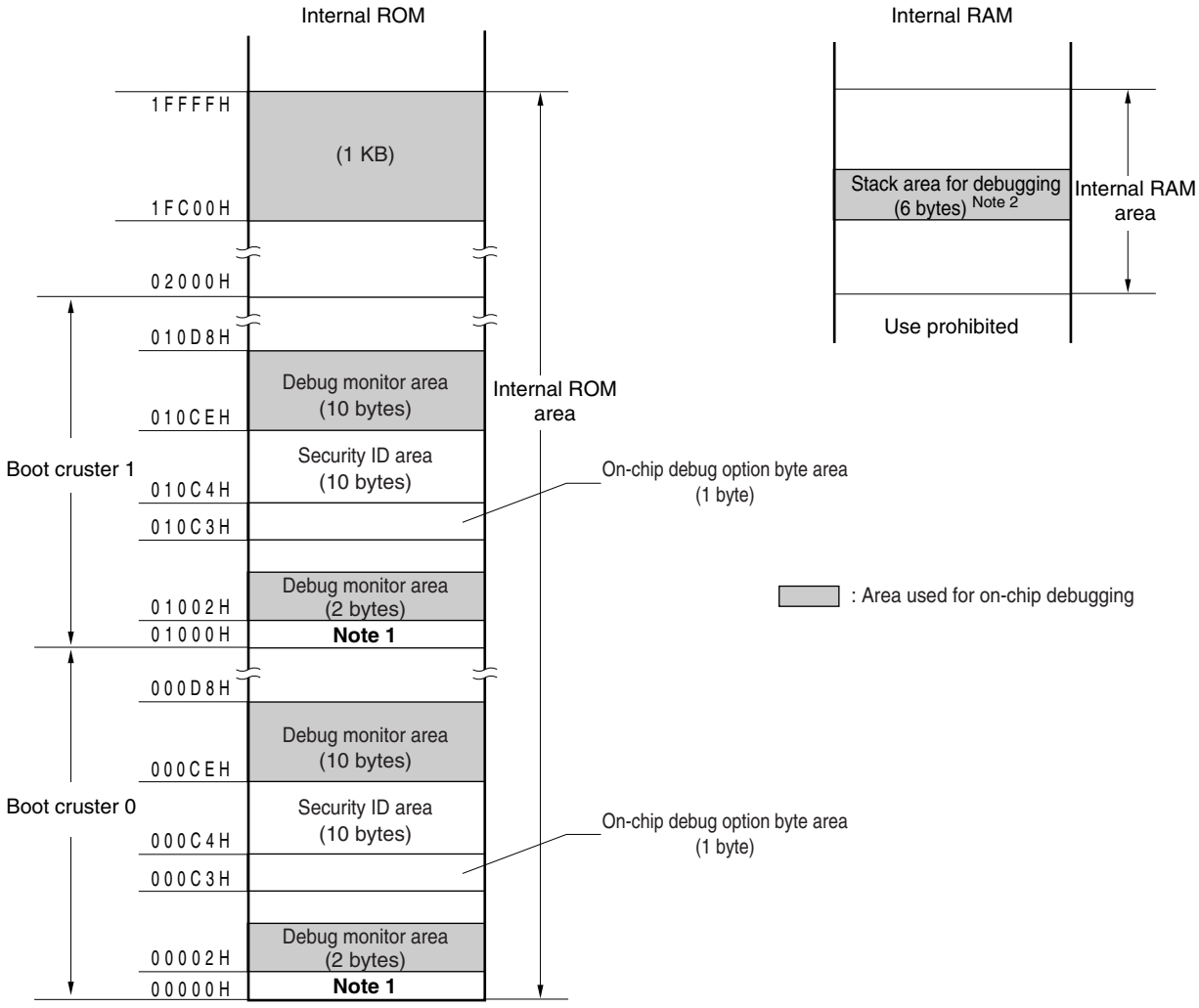
To perform communication between the 78K0R/Lx3-M microcontrollers and QB-MINI2, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler RA78K0R or compiler CC78K0R is used, the items can be set by using linker 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. In debugging, reset vector is rewritten to address allocated to a monitor program.

2. 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, 6 extra bytes are consumed for the stack area used.

For details of the way to secure of the memory space, refer to the **QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual (U18371E)**.

CHAPTER 30 BCD CORRECTION CIRCUIT

30.1 BCD Correction Circuit Function

The BCD correction circuit is mounted onto all 78K0R/Lx3-M microcontroller products.

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 BCDADJ register.

30.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

(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.

BCDADJ 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)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
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 BCDADJ register.

<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 register.

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 BCDADJ register.

<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 register.

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 78K0R/Lx3-M microcontroller instruction set. For details of each operation and operation code, refer to the separate document **78K0R Microcontrollers Instructions User's Manual (U17792E)**.

Remark The shaded parts of the tables in **Table 31-5 Operation List** indicate the operation or instruction format that is newly added for the 78K0R microcontrollers.

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)
sfrp	Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only[Note])
saddr	FFE20H to FFF1FH Immediate data or labels
saddrp	FFE20H to FF1FH Immediate data or labels (even addresses only[Note])
addr20	00000H to FFFFFH Immediate data or labels
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions[Note])
addr5	0080H to 00BFH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

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_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
∨	Exclusive logical sum (exclusive OR)
¯	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

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.

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

Remark The shaded parts of the tables in **Table 31-5 Operation List** indicate the operation or instruction format that is newly added for the 78K0R microcontrollers.

Table 31-5. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	–	r ← byte			
		saddr, #byte	3	1	–	(saddr) ← byte			
		sfr, #byte	3	1	–	sfr ← byte			
		!addr16, #byte	4	1	–	(addr16) ← byte			
		A, r Note 3	1	1	–	A ← r			
		r, A Note 3	1	1	–	r ← A			
		A, saddr	2	1	–	A ← (saddr)			
		saddr, A	2	1	–	(saddr) ← A			
		A, sfr	2	1	–	A ← sfr			
		sfr, A	2	1	–	sfr ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		!addr16, A	3	1	–	(addr16) ← A			
		PSW, #byte	3	3	–	PSW ← byte	×	×	×
		A, PSW	2	1	–	A ← PSW			
		PSW, A	2	3	–	PSW ← A	×	×	×
		ES, #byte	2	1	–	ES ← byte			
		ES, saddr	3	1	–	ES ← (saddr)			
		A, ES	2	1	–	A ← ES			
		ES, A	2	1	–	ES ← A			
		CS, #byte	3	1	–	CS ← byte			
		A, CS	2	1	–	A ← CS			
		CS, A	2	1	–	CS ← A			
		A, [DE]	1	1	4	A ← (DE)			
		[DE], A	1	1	–	(DE) ← A			
		[DE + byte], #byte	3	1	–	(DE + byte) ← byte			
		A, [DE + byte]	2	1	4	A ← (DE + byte)			
		[DE + byte], A	2	1	–	(DE + byte) ← A			
		A, [HL]	1	1	4	A ← (HL)			
		[HL], A	1	1	–	(HL) ← A			
		[HL + byte], #byte	3	1	–	(HL + byte) ← byte			

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (2/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	A, [HL + byte]	2	1	4	A ← (HL + byte)	
		[HL + byte], A	2	1	–	(HL + byte) ← A	
		A, [HL + B]	2	1	4	A ← (HL + B)	
		[HL + B], A	2	1	–	(HL + B) ← A	
		A, [HL + C]	2	1	4	A ← (HL + C)	
		[HL + C], A	2	1	–	(HL + C) ← A	
		word[B], #byte	4	1	–	(B + word) ← byte	
		A, word[B]	3	1	4	A ← (B + word)	
		word[B], A	3	1	–	(B + word) ← A	
		word[C], #byte	4	1	–	(C + word) ← byte	
		A, word[C]	3	1	4	A ← (C + word)	
		word[C], A	3	1	–	(C + word) ← A	
		word[BC], #byte	4	1	–	(BC + word) ← byte	
		A, word[BC]	3	1	4	A ← (BC + word)	
		word[BC], A	3	1	–	(BC + word) ← A	
		[SP + byte], #byte	3	1	–	(SP + byte) ← byte	
		A, [SP + byte]	2	1	–	A ← (SP + byte)	
		[SP + byte], A	2	1	–	(SP + byte) ← A	
		B, saddr	2	1	–	B ← (saddr)	
		B, !addr16	3	1	4	B ← (addr16)	
		C, saddr	2	1	–	C ← (saddr)	
		C, !addr16	3	1	4	C ← (addr16)	
		X, saddr	2	1	–	X ← (saddr)	
		X, !addr16	3	1	4	X ← (addr16)	
		ES:!addr16, #byte	5	2	–	(ES, addr16) ← byte	
		A, ES:!addr16	4	2	5	A ← (ES, addr16)	
		ES:!addr16, A	4	2	–	(ES, addr16) ← A	
		A, ES:[DE]	2	2	5	A ← (ES, DE)	
		ES:[DE], A	2	2	–	(ES, DE) ← A	
		ES:[DE + byte],#byte	4	2	–	((ES, DE) + byte) ← byte	
		A, ES:[DE + byte]	3	2	5	A ← ((ES, DE) + byte)	
		ES:[DE + byte], A	3	2	–	((ES, DE) + byte) ← A	

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (3/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	A, ES:[HL]	2	2	5	A ← (ES, HL)	
		ES:[HL], A	2	2	–	(ES, HL) ← A	
		ES:[HL + byte],#byte	4	2	–	((ES, HL) + byte) ← byte	
		A, ES:[HL + byte]	3	2	5	A ← ((ES, HL) + byte)	
		ES:[HL + byte], A	3	2	–	((ES, HL) + byte) ← A	
		A, ES:[HL + B]	3	2	5	A ← ((ES, HL) + B)	
		ES:[HL + B], A	3	2	–	((ES, HL) + B) ← A	
		A, ES:[HL + C]	3	2	5	A ← ((ES, HL) + C)	
		ES:[HL + C], A	3	2	–	((ES, HL) + C) ← A	
		ES:word[B], #byte	5	2	–	((ES, B) + word) ← byte	
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)	
		ES:word[B], A	4	2	–	((ES, B) + word) ← A	
		ES:word[C], #byte	5	2	–	((ES, C) + word) ← byte	
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)	
		ES:word[C], A	4	2	–	((ES, C) + word) ← A	
		ES:word[BC], #byte	5	2	–	((ES, BC) + word) ← byte	
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)	
		ES:word[BC], A	4	2	–	((ES, BC) + word) ← A	
		B, ES:!addr16	4	2	5	B ← (ES, addr16)	
		C, ES:!addr16	4	2	5	C ← (ES, addr16)	
		X, ES:!addr16	4	2	5	X ← (ES, addr16)	
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	–	A ←→ r	
		A, saddr	3	2	–	A ←→ (saddr)	
		A, sfr	3	2	–	A ←→ sfr	
		A, !addr16	4	2	–	A ←→ (addr16)	
		A, [DE]	2	2	–	A ←→ (DE)	
		A, [DE + byte]	3	2	–	A ←→ (DE + byte)	
		A, [HL]	2	2	–	A ←→ (HL)	
		A, [HL + byte]	3	2	–	A ←→ (HL + byte)	
		A, [HL + B]	2	2	–	A ←→ (HL + B)	
		A, [HL + C]	2	2	–	A ←→ (HL + C)	

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (4/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	XCH	A, ES:!addr16	5	3	–	A ←→ (ES, addr16)			
		A, ES:[DE]	3	3	–	A ←→ (ES, DE)			
		A, ES:[DE + byte]	4	3	–	A ←→ ((ES, DE) + byte)			
		A, ES:[HL]	3	3	–	A ←→ (ES, HL)			
		A, ES:[HL + byte]	4	3	–	A ←→ ((ES, HL) + byte)			
		A, ES:[HL + B]	3	3	–	A ←→ ((ES, HL) + B)			
		A, ES:[HL + C]	3	3	–	A ←→ ((ES, HL) + C)			
	ONEB	A	1	1	–	A ← 01H			
		X	1	1	–	X ← 01H			
		B	1	1	–	B ← 01H			
		C	1	1	–	C ← 01H			
		saddr	2	1	–	(saddr) ← 01H			
		!addr16	3	1	–	(addr16) ← 01H			
		ES:!addr16	4	2	–	(ES, addr16) ← 01H			
	CLRB	A	1	1	–	A ← 00H			
		X	1	1	–	X ← 00H			
		B	1	1	–	B ← 00H			
		C	1	1	–	C ← 00H			
		saddr	2	1	–	(saddr) ← 00H			
		!addr16	3	1	–	(addr16) ← 00H			
		ES:!addr16	4	2	–	(ES,addr16) ← 00H			
	MOVS	[HL + byte], X	3	1	–	(HL + byte) ← X	×		×
		ES:[HL + byte], X	4	2	–	(ES, HL + byte) ← X	×		×
16-bit data transfer	MOVW	rp, #word	3	1	–	rp ← word			
		saddrp, #word	4	1	–	(saddrp) ← word			
		sfrp, #word	4	1	–	sfrp ← word			
		AX, saddrp	2	1	–	AX ← (saddrp)			
		saddrp, AX	2	1	–	(saddrp) ← AX			
		AX, sfrp	2	1	–	AX ← sfrp			
		sfrp, AX	2	1	–	sfrp ← AX			
		AX, rp Note 3	1	1	–	AX ← rp			
		rp, AX Note 3	1	1	–	rp ← AX			

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except rp = AX

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (5/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, !addr16	3	1	4	AX ← (addr16)			
		!addr16, AX	3	1	–	(addr16) ← AX			
		AX, [DE]	1	1	4	AX ← (DE)			
		[DE], AX	1	1	–	(DE) ← AX			
		AX, [DE + byte]	2	1	4	AX ← (DE + byte)			
		[DE + byte], AX	2	1	–	(DE + byte) ← AX			
		AX, [HL]	1	1	4	AX ← (HL)			
		[HL], AX	1	1	–	(HL) ← AX			
		AX, [HL + byte]	2	1	4	AX ← (HL + byte)			
		[HL + byte], AX	2	1	–	(HL + byte) ← AX			
		AX, word[B]	3	1	4	AX ← (B + word)			
		word[B], AX	3	1	–	(B + word) ← AX			
		AX, word[C]	3	1	4	AX ← (C + word)			
		word[C], AX	3	1	–	(C + word) ← AX			
		AX, word[BC]	3	1	4	AX ← (BC + word)			
		word[BC], AX	3	1	–	(BC + word) ← AX			
		AX, [SP + byte]	2	1	–	AX ← (SP + byte)			
		[SP + byte], AX	2	1	–	(SP + byte) ← AX			
		BC, saddrp	2	1	–	BC ← (saddrp)			
		BC, !addr16	3	1	4	BC ← (addr16)			
		DE, saddrp	2	1	–	DE ← (saddrp)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		HL, saddrp	2	1	–	HL ← (saddrp)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)			
		ES:!addr16, AX	4	2	–	(ES, addr16) ← AX			
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)			
		ES:[DE], AX	2	2	–	(ES, DE) ← AX			
		AX, ES:[DE + byte]	3	2	5	AX ← ((ES, DE) + byte)			
		ES:[DE + byte], AX	3	2	–	((ES, DE) + byte) ← AX			
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)			
		ES:[HL], AX	2	2	–	(ES, HL) ← AX			

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (6/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, ES:[HL + byte]	3	2	5	AX ← ((ES, HL) + byte)			
		ES:[HL + byte], AX	3	2	–	((ES, HL) + byte) ← AX			
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	–	((ES, B) + word) ← AX			
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	–	((ES, C) + word) ← AX			
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	–	((ES, BC) + word) ← AX			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
	XCHW	AX, rp Note 3	1	1	–	AX ←→ rp			
	ONEW	AX	1	1	–	AX ← 0001H			
		BC	1	1	–	BC ← 0001H			
	CLRW	AX	1	1	–	AX ← 0000H			
		BC	1	1	–	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) + byte	×	×	×
		A, r Note 4	2	1	–	A, CY ← A + r	×	×	×
		r, A	2	1	–	r, CY ← r + A	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A,CY ← A + ((ES, HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A,CY ← A + ((ES, HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A,CY ← A + ((ES, HL) + C)	×	×	×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except rp = AX

4. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (7/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	ADDC	A, #byte	2	1	–	A, CY ← A + byte + CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A + r + CY	×	×	×
		r, A	2	1	–	r, CY ← r + A + CY	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr) + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C) + CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16) + CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A + (ES, HL) + CY	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A + ((ES, HL) + byte) + CY	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A + ((ES, HL) + B) + CY	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A + ((ES, HL) + C) + CY	×	×	×
	SUB	A, #byte	2	1	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r	×	×	×
		r, A	2	1	–	r, CY ← r – A	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A – (HL + C)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A – ((ES:HL) + C)	×	×	×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (8/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r – CY	×	×	×
		r, A	2	1	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A – (HL + C) – CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES:addr16) – CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A – (ES:HL) – CY	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A – ((ES:HL) + byte) – CY	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A – ((ES:HL) + B) – CY	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A – ((ES:HL) + C) – CY	×	×	×
	AND	A, #byte	2	1	–	A ← A ∧ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	–	A ← A ∧ r	×		
		r, A	2	1	–	r ← r ∧ A	×		
		A, saddr	2	1	–	A ← A ∧ (saddr)	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	1	4	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	1	4	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	1	4	A ← A ∧ (HL + C)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	A ← A ∧ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	A ← A ∧ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	A ← A ∧ ((ES:HL) + C)	×		

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (9/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	OR	A, #byte	2	1	–	A ← A ∨ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∨ byte	×		
		A, r Note 3	2	1	–	A ← A ∨ r	×		
		r, A	2	1	–	r ← r ∨ A	×		
		A, saddr	2	1	–	A ← A ∨ (saddr)	×		
		A, !addr16	3	1	4	A ← A ∨ (addr16)	×		
		A, [HL]	1	1	4	A ← A ∨ (HL)	×		
		A, [HL + byte]	2	1	4	A ← A ∨ (HL + byte)	×		
		A, [HL + B]	2	1	4	A ← A ∨ (HL + B)	×		
		A, [HL + C]	2	1	4	A ← A ∨ (HL + C)	×		
		A, ES:!addr16	4	2	5	A ← A ∨ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	A ← A ∨ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	A ← A ∨ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	A ← A ∨ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	A ← A ∨ ((ES:HL) + C)	×		
	XOR	A, #byte	2	1	–	A ← A ∨̲ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∨̲ byte	×		
		A, r Note 3	2	1	–	A ← A ∨̲ r	×		
		r, A	2	1	–	r ← r ∨̲ A	×		
		A, saddr	2	1	–	A ← A ∨̲ (saddr)	×		
		A, !addr16	3	1	4	A ← A ∨̲ (addr16)	×		
		A, [HL]	1	1	4	A ← A ∨̲ (HL)	×		
		A, [HL + byte]	2	1	4	A ← A ∨̲ (HL + byte)	×		
		A, [HL + B]	2	1	4	A ← A ∨̲ (HL + B)	×		
		A, [HL + C]	2	1	4	A ← A ∨̲ (HL + C)	×		
		A, ES:!addr16	4	2	5	A ← A ∨̲ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	A ← A ∨̲ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	A ← A ∨̲ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	A ← A ∨̲ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	A ← A ∨̲ ((ES:HL) + C)	×		

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (10/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	–	A – byte	×	×	×
		saddr, #byte	3	1	–	(saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A – r	×	×	×
		r, A	2	1	–	r – A	×	×	×
		A, saddr	2	1	–	A – (saddr)	×	×	×
		A, !addr16	3	1	4	A – (addr16)	×	×	×
		A, [HL]	1	1	4	A – (HL)	×	×	×
		A, [HL + byte]	2	1	4	A – (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A – (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A – (HL + C)	×	×	×
		!addr16, #byte	4	1	4	(addr16) – byte	×	×	×
		A, ES:!addr16	4	2	5	A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A – ((ES:HL) + C)	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	×	×	×
<R>	CMP0	A	1	1	–	A – 00H	×	0	0
		X	1	1	–	X – 00H	×	0	0
		B	1	1	–	B – 00H	×	0	0
		C	1	1	–	C – 00H	×	0	0
		saddr	2	1	–	(saddr) – 00H	×	0	0
		!addr16	3	1	4	(addr16) – 00H	×	0	0
		ES:!addr16	4	2	5	(ES:addr16) – 00H	×	0	0
	CMPS	X, [HL + byte]	3	1	4	X – (HL + byte)	×	×	×
		X, ES:[HL + byte]	4	2	5	X – ((ES:HL) + byte)	×	×	×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (11/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	1	–	AX, CY ← AX + word	×	×	×
		AX, AX	1	1	–	AX, CY ← AX + AX	×	×	×
		AX, BC	1	1	–	AX, CY ← AX + BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX + DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX + HL	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX + (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX + (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX + (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX + (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX + ((ES:HL) + byte)	×	×	×
	SUBW	AX, #word	3	1	–	AX, CY ← AX – word	×	×	×
		AX, BC	1	1	–	AX, CY ← AX – BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX – DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX – HL	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX – (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX – (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX – (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX – ((ES:HL) + byte)	×	×	×
	CMPW	AX, #word	3	1	–	AX – word	×	×	×
		AX, BC	1	1	–	AX – BC	×	×	×
		AX, DE	1	1	–	AX – DE	×	×	×
		AX, HL	1	1	–	AX – HL	×	×	×
		AX, saddrp	2	1	–	AX – (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX – (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX – (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX – ((ES:HL) + byte)	×	×	×
Multiply	MULU	X	1	1	–	AX ← A × X			

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (12/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Increment/ decrement	INC	r	1	1	–	r ← r + 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) + 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) + 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) + 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) + 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) + 1	×	×	
	DEC	r	1	1	–	r ← r – 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) – 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) – 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) – 1	×	×	
	INCW	rp	1	1	–	rp ← rp + 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) + 1			
		!addr16	3	2	–	(addr16) ← (addr16) + 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) + 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) + 1			
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) + 1			
	DECW	rp	1	1	–	rp ← rp – 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) – 1			
		!addr16	3	2	–	(addr16) ← (addr16) – 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1			
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) – 1			
Shift	SHR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) × cnt			×
	SHRW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) × cnt			×
	SHL	A, cnt	2	1	–	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) × cnt			×
		B, cnt	2	1	–	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) × cnt			×
		C, cnt	2	1	–	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) × cnt			×
	SHLW	AX, cnt	2	1	–	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) × cnt			×
		BC, cnt	2	1	–	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) × cnt			×
	SAR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) × cnt			×
	SARW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← AX_{15}) × cnt			×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

3. cnt indicates the bit shift count.

Table 31-5. Operation List (13/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Rotate	ROR	A, 1	2	1	–	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1			×
	ROL	A, 1	2	1	–	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1			×
	RORC	A, 1	2	1	–	(CY $\leftarrow A_0$, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1			×
	ROLC	A, 1	2	1	–	(CY $\leftarrow A_7$, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1			×
	ROLWC	AX,1	2	1	–	(CY $\leftarrow AX_{15}$, $AX_0 \leftarrow$ CY, $AX_{m+1} \leftarrow AX_m$) × 1			×
		BC,1	2	1	–	(CY $\leftarrow BC_{15}$, $BC_0 \leftarrow$ CY, $BC_{m+1} \leftarrow BC_m$) × 1			×
Bit manipulate	MOV1	CY, saddr.bit	3	1	–	CY ← (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← sfr.bit			×
		CY, A.bit	2	1	–	CY ← A.bit			×
		CY, PSW.bit	3	1	–	CY ← PSW.bit			×
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		saddr.bit, CY	3	2	–	(saddr).bit ← CY			
		sfr.bit, CY	3	2	–	sfr.bit ← CY			
		A.bit, CY	2	1	–	A.bit ← CY			
		PSW.bit, CY	3	4	–	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	2	–	(HL).bit ← CY			
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit			×
		ES:[HL].bit, CY	3	3	–	(ES, HL).bit ← CY			
	AND1	CY, saddr.bit	3	1	–	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∧ PSW.bit			×
		CY,[HL].bit	2	1	4	CY ← CY ∧ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit			×
	OR1	CY, saddr.bit	3	1	–	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∨ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∨ PSW.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (14/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, saddr.bit	3	1	–	CY ← CY ⊻ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ⊻ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ⊻ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ⊻ PSW.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ⊻ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ⊻ (ES, HL).bit			×
	SET1	saddr.bit	3	2	–	(saddr).bit ← 1			
		sfr.bit	3	2	–	sfr.bit ← 1			
		A.bit	2	1	–	A.bit ← 1			
		!addr16.bit	4	2	–	(addr16).bit ← 1			
		PSW.bit	3	4	–	PSW.bit ← 1	×	×	×
		[HL].bit	2	2	–	(HL).bit ← 1			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 1			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 1			
	CLR1	saddr.bit	3	2	–	(saddr.bit) ← 0			
		sfr.bit	3	2	–	sfr.bit ← 0			
		A.bit	2	1	–	A.bit ← 0			
		!addr16.bit	4	2	–	(addr16).bit ← 0			
		PSW.bit	3	4	–	PSW.bit ← 0	×	×	×
		[HL].bit	2	2	–	(HL).bit ← 0			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 0			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 0			
	SET1	CY	2	1	–	CY ← 1			1
	CLR1	CY	2	1	–	CY ← 0			0
	NOT1	CY	2	1	–	CY ← $\overline{CY}$			×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (15/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/ return	CALL	rp	2	3	–	(SP – 2) ← (PC + 2)S, (SP – 3) ← (PC + 2)H, (SP – 4) ← (PC + 2)L, PC ← CS, rp, SP ← SP – 4			
		$!addr20	3	3	–	(SP – 2) ← (PC + 3)S, (SP – 3) ← (PC + 3)H, (SP – 4) ← (PC + 3)L, PC ← PC + 3 + jdisp16, SP ← SP – 4			
		!addr16	3	3	–	(SP – 2) ← (PC + 3)S, (SP – 3) ← (PC + 3)H, (SP – 4) ← (PC + 3)L, PC ← 0000, addr16, SP ← SP – 4			
		!!addr20	4	3	–	(SP – 2) ← (PC + 4)S, (SP – 3) ← (PC + 4)H, (SP – 4) ← (PC + 4)L, PC ← addr20, SP ← SP – 4			
	CALLT	[addr5]	2	5	–	(SP – 2) ← (PC + 2)S, (SP – 3) ← (PC + 2)H, (SP – 4) ← (PC + 2)L , PCS ← 0000, PCH ← (0000, addr5 + 1), PCL ← (0000, addr5), SP ← SP – 4			
	BRK	–	2	5	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 2)S, (SP – 3) ← (PC + 2)H, (SP – 4) ← (PC + 2)L, PCS ← 0000, PCH ← (0007FH), PCL ← (0007EH), SP ← SP – 4, IE ← 0			
	RET	–	1	6	–	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), SP ← SP + 4			
	RETI	–	2	6	–	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R
	RETB	–	2	6	–	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulate	PUSH	PSW	2	1	–	(SP – 1) ← PSW, (SP – 2) ← 00H, SP ← SP – 2			
		rp	1	1	–	(SP – 1) ← rp_H, (SP – 2)← rp_L, SP ← SP – 2			
	POP	PSW	2	3	–	PSW ← (SP + 1), SP ← SP + 2	R	R	R
		rp	1	1	–	rp_L ← (SP), rp_H ← (SP + 1), SP ← SP + 2			
	MOVW	SP, #word	4	1	–	SP ← word			
		SP, AX	2	1	–	SP ← AX			
		AX, SP	2	1	–	AX ← SP			
		HL, SP	3	1	–	HL ← SP			
		BC, SP	3	1	–	BC ← SP			
		DE, SP	3	1	–	DE ← SP			
	ADDW	SP, #byte	2	1	–	SP ← SP + byte			
	SUBW	SP, #byte	2	1	–	SP ← SP – byte			
Unconditional branch	BR	AX	2	3	–	PC ← CS, AX			
		$addr20	2	3	–	PC ← PC + 2 + jdisp8			
		$!addr20	3	3	–	PC ← PC + 3 + jdisp16			
		!addr16	3	3	–	PC ← 0000, addr16			
		!!addr20	4	3	–	PC ← addr20			
Conditional branch	BC	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4[Note 3]	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=0			
	BNH	$addr20	3	2/4[Note 3]	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=1			
	BT	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5[Note 3]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 31-5. Operation List (17/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5[Note 3]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6[Note 3]	–	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	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. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

3. n indicates the number of register banks (n = 0 to 3)

CHAPTER 32 ELECTRICAL SPECIFICATIONS

Cautions 1. The 78K0R/Lx3-M microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. The pins mounted depend on the product. Refer to 1.3 Pin Configuration (Top View) and CHAPTER 2 PIN FUNCTIONS.

Absolute Maximum Ratings (T_A = 25°C) (1/3)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.5 to +4.6	V
	LV_{DD}		−0.5 to +4.6	V
	V_{SS}		−0.5 to +0.3	V
	LV_{SS}		−0.5 to +0.3	V
	AV_{REFIN} Note 1		−0.5 to LAV_{DD} +0.5 Note 2	V
	AV_{DD}		−0.5 to V_{DD} +0.3 Note 2	V
	LAV_{DD}		−0.5 to LV_{DD} +0.5 Note 2	V
	LAV_{SS}		−0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC	−0.3 to +3.6 and −0.3 to V_{DD} +0.3 Note 3	V
Input voltage	V_{I1}	P00, P01, P10 to P12, P33, P40, P41, P50 to P57, P81, P82, P90 to P97, P100, P120 to P122, P124, P140 to P147, EXCLK, $\overline{RESET}$, FLMD0	−0.3 to V_{DD} +0.3 Note 1	V
	V_{I2}	P60, P61 (N-ch open-drain)	−0.3 to +6.5	V
	V_{I3}	P152, P157	−0.3 to AV_{DD} +0.3 and −0.3 to V_{DD} +0.3 Note 2	V
	V_{I4}	LP01, LRESET, XT1, IC3	−0.3 to LV_{DD} +0.3 Note 2	V

Notes 1. Applied voltage of AV_{REFIO} pin.

2. Must be 4.6 V or lower.

3. Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

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 port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/3)

Parameter	Symbols	Conditions		Ratings	Unit
Output voltage	V_{O1}	P00, P01, P10 to P12, P33, P40, P41, P50 to P57, P60, P61, P81, P82, P90 to P97, P100, P120, P140 to P147		−0.3 to V_{DD} +0.3[Note]	V
	V_{O2}	P152, P157		−0.3 to AV_{DD} +0.3[Note]	V
	V_{O3}	SEG0 to SEG39, COM0 to COM3	External resistance division method, capacitor split method	−0.3 to V_{DD} +0.3[Note]	V
			Internal voltage boosting method	−0.3 to V_{LC0} +0.3[Note]	V
	V_{O4}	LP01, P16, CF, RTC1HZ, FXTOUT		−0.3 to LV_{DD} +0.3[Note]	V
Analog input voltage	V_{AI1}	ANI10, ANI15		−0.3 to AV_{DD} +0.3 and −0.3 to V_{DD} +0.3[Note]	V
	V_{AI2}	ANIDS00, ANIDS01, ANIDS10, ANIDS11, ANIDS20, ANIDS21, ANIDS30, ANIDS31		−0.5 to LAV_{DD} +0.5[Note]	V
Analog input reference voltage	AV_{REF}			−0.3 to AV_{DD} +0.3[Note]	V
Output current, high	I_{OH1}	Per pin	P00, P01, P10 to P12, P33, P40, P41, P81, P82, P120	−10	mA
			P50 to P57, P90 to P97, P100, P140 to P147	−10	mA
			LP01, P16, CF, RTC1HZ, FXTOUT	−4	mA
		Total of all pins −85 mA	P00, P01, P10 to P12, P33, P40, P41, P81, P82, P120	−25	mA
			P50 to P57, P90 to P97, P100, P140 to P147	−25	mA
			LP01, P16, CF, RTC1HZ, FXTOUT	−35	mA
	I_{OH2}	Per pin	P152, P157	−0.5	mA
		Total of all pins		−2	mA

Note Must be 4.6 V or lower.

Cautions 1. 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.

2. The value of the current that can be run per pin must satisfy the value of the current per pin and the total value of the currents of all pins.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (3/3)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, low	I_{OL1}	Per pin	P00, P01, P10 to P12, P33, P40, P41, P81, P82, P120	30	mA
			P60, P61	30	mA
			P50 to P57, P90 to P97, P100, P140 to P147	10	mA
			LP01, P16, CF, RTC1HZ, FXTOUT	4	mA
		Total of all pins 200 mA	P00, P01, P10 to P12, P33, P40, P41, P81, P82, P120	80	mA
			P60, P61	60	mA
			P50 to P57, P90 to P97, P100, P140 to P147	25	mA
			LP01, P16, CF, RTC1HZ, FXTOUT	35	mA
	I_{OL2}	Per pin	P152, P157	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A	In normal operation mode		−40 to +85	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			−65 to +150	°C

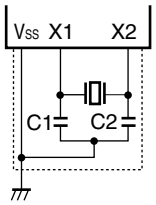
Cautions 1. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

2. The value of the current that can be run per pin must satisfy the value of the current per pin and the total value of the currents of all pins.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

X1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = AV_{DD} = LV_{DD} = LAV_{DD} ≤ 3.6 V, V_{SS} = LV_{SS} = LAV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator, crystal resonator		X1 clock oscillation frequency (f_X)[Note]	2.7 V ≤ V_{DD} ≤ 3.6 V	2.0		20.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V	2.0		5.0	

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. Since the CPU is started by the internal high-speed oscillation 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 oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Internal Oscillator Characteristics

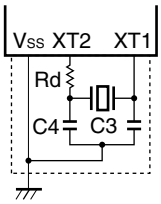
(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = AV_{DD} = LV_{DD} = LAV_{DD} ≤ 3.6 V, V_{SS} = LV_{SS} = LAV_{SS} = 0 V)

Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
Internal high-speed oscillation clock frequency (f_{IH})[Note]	f_{IH1M}	Low-power consumption mode		0.87	1	1.13	MHz
	f_{IH8M}			7.6	8	8.4	MHz
	f_{IH20M}	2.7 V ≤ V_{DD} ≤ 3.6 V		19	20	21	MHz
Internal low-speed oscillation clock frequency (f_{IL})	f_{IL}	Normal power mode	2.7 V ≤ V_{DD} ≤ 3.6 V	24	30	36	kHz
			1.8 V ≤ V_{DD} < 2.7 V	21	30	39	kHz
		Low-power consumption mode		21	30	39	kHz

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

XT1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = AV_{DD} = LV_{DD} = LAV_{DD} ≤ 3.6 V, V_{SS} = LV_{SS} = LAV_{SS} = 0 V)

Resonator	Recommended Circuit	Items	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency (f_{XT})Note		31	32.768	39	kHz

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Cautions 1. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (1/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = AV_{DD} = LV_{DD} = LAV_{DD} ≤ 3.6 V, V_{SS} = LV_{SS} = LAV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00, P01, P10 to P12, P33, P40, P41, P81, P82, P120				–1.0	mA
		Per pin for P50 to P57, P90 to P97, P100, P140 to P147				–0.45	mA
		Total of P00, P01, P10 to P12, P33, P40, P41, P81, P82, P120 (When duty = 70%[Note 2])	2.7 V ≤ V_{DD} ≤ 3.6 V			–10.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			–5.0	mA
		Total of P50 to P57, P90 to P97, P100, P140 to P147 (When duty = 70%[Note 2])				–3.6	mA
		Total of all pins (When duty = 60%[Note 2])	2.7 V ≤ V_{DD} ≤ 3.6 V			–13.6	mA
			1.8 V ≤ V_{DD} < 2.7 V			–8.6	mA
	I_{OH2}	Per pin for P152, P157				–0.1	mA
	I_{OH3}	Per pin for LP01, P16, CF, RTC1HZ, FXTOUT	2.7 V ≤ V_{DD} ≤ 3.6 V			–2.5	mA
			1.8 V ≤ V_{DD} < 2.7 V			–1.0	mA
		Total of LP01, P16, CF, RTC1HZ, FXTOUT	2.7 V ≤ V_{DD} ≤ 3.6 V			–10	mA
			1.8 V ≤ V_{DD} < 2.7 V			–5.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} pin to an output pin.

2. Specification under conditions where the duty factor is 60% or 70%.

The output current value that has changed 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_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 50% and I_{OH} = –20.0 mA

Total output current of pins = (–20.0 × 0.7)/(50 × 0.01) = –28.0 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 P10 to P12, and P82 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 port pins.

DC Characteristics (2/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = AV_{DD} = LV_{DD} = LAV_{DD} ≤ 3.6 V, V_{SS} = LV_{SS} = LAV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P00, P01, P12, P33, P40, P41, P81, P82, P120	2.7 V ≤ V_{DD} ≤ 3.6 V			1.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			0.5	mA
		Per pin for P10, P11	2.7 V ≤ V_{DD} ≤ 3.6 V			1.5	mA
			1.8 V ≤ V_{DD} < 2.7 V			0.6	mA
		Per pin for P60, P61	2.7 V ≤ V_{DD} ≤ 3.6 V			3.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			2.0	mA
		Per pin for P50 to P57, P90 to P97, P100, P140 to P147	2.7 V ≤ V_{DD} ≤ 3.6 V			0.8	mA
			1.8 V ≤ V_{DD} < 2.7 V			0.35	mA
		Total of P00, P01, P10 to P12, P33, P40, P41, P81, P82, P120 (When duty = 70%[Note 2])	2.7 V ≤ V_{DD} ≤ 3.6 V			15.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			9.0	mA
		Total of P60, P61 (When duty = 70%[Note 2])	2.7 V ≤ V_{DD} ≤ 3.6 V			6.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			4.0	mA
		Total of P50 to P57, P90 to P97, P100, P140 to P147 (When duty = 70%[Note 2])	2.7 V ≤ V_{DD} ≤ 3.6 V			6.4	mA
			1.8 V ≤ V_{DD} < 2.7 V			2.8	mA
		Total of all pins (When duty = 70%[Note 2])	2.7 V ≤ V_{DD} ≤ 3.6 V			27.4	mA
			1.8 V ≤ V_{DD} < 2.7 V			15.8	mA
	I_{OL2}	Per pin for P152, P157				0.4	mA
	I_{OL3}	Per pin for LP01, P16, CF, RTC1HZ, FXTOUT	2.7 V ≤ V_{DD} ≤ 3.6 V			2.5	mA
			1.8 V ≤ V_{DD} < 2.7 V			1.0	mA
		Total of LP01, P16, CF, RTC1HZ, FXTOUT	2.7 V ≤ V_{DD} ≤ 3.6 V			15	mA
			1.8 V ≤ V_{DD} < 2.7 V			9.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to V_{SS} and AV_{SS} pin.

2. Specification under conditions where the duty factor is 60% or 70%.

The output current value that has changed 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_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 50% and I_{OL} = 20.0 mA

Total output current of pins = (20.0 × 0.7)/(50 × 0.01) = 28.0 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 port pins.

DC Characteristics (3/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = AV_{DD} = LV_{DD} = LAV_{DD} ≤ 3.6 V, V_{SS} = LV_{SS} = LAV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00, P01, P12, P41, P51, P54 to P57, P82, P90 to P97, P100, P124, P140 to P147		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P10, P11, P33, P40, P50, P52, P53, P81, P120 to P122, $\overline{RESET}$	Normal input buffer	0.8V_{DD}		V_{DD}	V
	V_{IH3}	P10, P11	TTL input buffer 2.7 V ≤ V_{DD} ≤ 3.6 V	2.0		V_{DD}	V
			TTL input buffer 1.8 V ≤ V_{DD} < 2.7 V	1.6		V_{DD}	V
	V_{IH4}	P152, P157		0.7AV_{DD}		AV_{DD}	V
	V_{IH5}	P60, P61		0.7V_{DD}		6.0	V
	V_{IH6}	FLMD0		0.9V_{DD} Note 1		V_{DD}	V
	V_{IH7}	EXCLKS0		0.8VDD		V_{DD}	V
	V_{IH8}	LP01		0.7LV_{DD}		LV_{DD}	V
	V_{IH9}	LRESET, EXCLKS1, IC3		0.8LV_{DD}		LV_{DD}	V
Input voltage, low	V_{IL1}	P00, P01, P12, P41, P51, P54 to P57, P82, P90 to P97, P100, P124, P140 to P147		0		0.3V_{DD}	V
	V_{IL2}	P10, P11, P33, P40, P50, P52, P53, P81, P120 to P122, $\overline{RESET}$	Normal input buffer	0		0.2V_{DD}	V
	V_{IL3}	P10, P11	TTL input buffer 2.7 V ≤ V_{DD} ≤ 3.6 V	0		0.5	V
			TTL input buffer 1.8 V ≤ V_{DD} < 2.7 V	0		0.2	V
	V_{IL4}	P152, P157		0		0.3AV_{DD}	V
	V_{IL5}	P60, P61		0		0.3V_{DD}	V
	V_{IL6}	FLMD0		0		0.1V_{DD} Note 2	V
	V_{IL7}	EXCLKS0		0		0.346	V
	V_{IL8}	LP01		0		0.3LV_{DD}	V
	V_{IL9}	LRESET, EXCLKS1, IC3		0		0.2LV_{DD}	V

Notes 1. Must be 0.9V_{DD} or higher when used in the flash memory programming mode.

2. When disabling writing of the flash memory, connect the FLMD0 pin directly to V_{SS}, and maintain a voltage less than 0.1V_{DD}.

Caution The maximum value of V_{IH} of pins P10 to P12, and P82 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 port pins.

DC Characteristics (4/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = AV_{DD} = LV_{DD} = LAV_{DD} ≤ 3.6 V, V_{SS} = LV_{SS} = LAV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00, P01, P10 to P12, P33, P40, P41, P81, P82, P120	I_{OH1} = –1.0 mA	V_{DD} –0.5			V
		P50 to P57, P90 to P97, P100, P140 to P147	I_{OH1} = –0.45 mA	V_{DD} –0.5			V
	V_{OH2}	P152, P157	I_{OH2} = –0.1 mA	AV_{DD} –0.5			V
	V_{OH3}	LP01, P16, CF, RTC1HZ	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OH3} = –2.5 mA	LV_{DD} –0.5			V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OH3} = –1.0 mA	LV_{DD} –0.5			V
	V_{OH4}	FXTOUT	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OH3} = –2.5 mA	0.8LV_{DD}			V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OH3} = –1.0 mA	0.8LV_{DD}			V
Output voltage, low	V_{OL1}	P00, P01, P12, P33, P40, P41, P81, P82, P120	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 1.0 mA			0.5	V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 0.5 mA			0.4	V
		P10, P11	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 1.5 mA			0.5	V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 0.6 mA			0.4	V
		P50 to P57, P90 to P97, P100, P140 to P147	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 0.8 mA			0.5	V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 0.35 mA			0.4	V
	V_{OL2}	P152, P157	I_{OL2} = 0.4 mA			0.4	V
	V_{OL3}	P60, P61	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 3.0 mA			0.4	V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 2.0 mA			0.4	V
	V_{OL4}	LP01, P16, CF, RTC1HZ	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL3} = 2.5 mA			0.4	V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OL3} = 1.0 mA			0.4	V
	V_{OL5}	FXTOUT	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL3} = 2.5 mA			0.346	V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OL3} = 1.0 mA			0.346	V

Caution P10 to P12, and P82 do not output high level in N-ch open-drain mode.

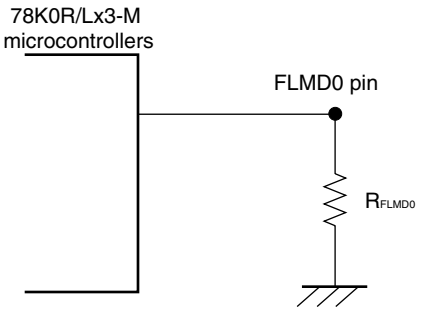
Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (5/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = AV_{DD} = LV_{DD} = LAV_{DD} ≤ 3.6 V, V_{SS} = LV_{SS} = LAV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00, P01, P10 to P12, P33, P40, P41, P50 to P57, P60, P61, P81, P82, P90 to P97, P100, P120, P140 to P147, FLMD0, RESET, IC0 to IC2, IC4 to IC6	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P152, P157	$V_I = AV_{DD}$				1	μA
	I_{LIH3}	P121, P122, P124 (X1, X2)	$V_I = V_{DD}$	In input port/ external clock			1	μA
				In resonator connection			10	μA
	I_{LIH4}	XT1	$V_I = LV_{DD}$	External clock			5	μA
	I_{LIH5}	LP01, LRESET, IC3	$V_I = LV_{DD}$				5	μA
Input leakage current, low	I_{LIL1}	P00, P01, P10 to P12, P33, P40, P41, P50 to P57, P60, P61, P81, P82, P90 to P97, P100, P120, P140 to P147, FLMD0, RESET, IC0 to IC2, IC4 to IC6	$V_I = V_{SS}$				–1	μA
	I_{LIL2}	P152, P157	$V_I = V_{SS}$				–1	μA
	I_{LIL3}	P121, P122, P124 (X1, X2)	$V_I = V_{SS}$	In input port/ external clock			–1	μA
				In resonator connection			–10	μA
	I_{LIL4}	XT1	$V_I = V_{SS}$	External clock			–5	μA
	I_{LIL5}	LP01, LRESET, IC3	$V_I = V_{SS}$				–5	μA
On-chip pull-up resistance	R_{U1}	P00, P01, P10 to P12, P33, P40, P41, P50 to P57, P81, P82, P90 to P97, P100, P120, P140 to P147	$V_I = V_{SS}$, in input port		10	20	100	kΩ
	R_{U2}	LP01, LRESET	$V_I = V_{SS}$		10	30	100	kΩ
FLMD0 pin external pull-down resistance[Note]	R_{FLMD0}	When enabling the self-programming mode setting with software			100			kΩ

Note It is recommended to leave the FLMD0 pin open. If the pin is required to be pulled down externally, set R_{FLMD0} to 100 kΩ or more.



Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (6/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = LV_{DD} ≤ 3.6 V, V_{SS} = LV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD1} Note 1	Operating mode	f_{MX} = 20 MHz, V_{DD} = 3.0 V Note 2	Square wave input		5.5	7.7	mA
				Resonator connection		5.8	8.0	
			f_{MX} = 10 MHz, V_{DD} = 3.0 V Notes 2, 3	Square wave input		3.2	4.6	mA
				Resonator connection		3.3	4.7	
			f_{MX} = 5 MHz, V_{DD} = 3.0 V Notes 2, 3	Square wave input		1.8	2.7	mA
				Resonator connection		1.9	2.8	
			f_{MX} = 5 MHz, V_{DD} = 2.0 V Notes 2, 3	Square wave input		1.3	2.2	mA
				Resonator connection		1.3	2.2	
			f_{IH} = 20 MHz Note 4	V_{DD} = 3.0 V		5.7	8.0	mA
			f_{IH} = 8 MHz Note 4	V_{DD} = 3.0 V		2.6	3.7	mA
			f_{IH} = 1 MHz, RMC = 5AH, OSMC = 02H Note 4	V_{DD} = 3.0 V		190	354	μA
			f_{SUB} = 32.768 kHz, FSEL = 0, SDIV = 1 Note 5	T_A = –40 to +50 °C	V_{DD} current		3.9	8.4	μA
					LV_{DD} current			15.0	
				T_A = –40 to +70 °C	V_{DD} current		3.9	11.3	μA
					LV_{DD} current			31.0	
				T_A = –40 to +85 °C	V_{DD} current		3.9	14.6	μA
					LV_{DD} current		25	47	

<R> **Notes 1.** Total current flowing into V_{DD}, AV_{DD}, LAV_{DD}, LV_{DD}, V_{LC0} to V_{LC2}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}, and excluding the current flowing into the watchdog timer, LVI circuit, A/D converter, LCD controller/driver, power calculation circuit, I/O port, and on-chip pull-up/pull-down resistors. The maximum values include the peripheral operation current.

2. When internal high-speed oscillator and subsystem clock are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0 and FLPC, FSEL (bits 1, 0 of operation speed mode control register (OSMC)) = 0, 0.

4. When high-speed system clock and subsystem clock are stopped.

5. When internal high-speed oscillation, and high-speed system clock are stopped. When watchdog timer is <R> stopped. When real-time counter (RTC) is operating.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

DC Characteristics (7/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = LV_{DD} ≤ 3.6 V, V_{SS} = LV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD2} Note 1	HALT mode	f_{MX} = 20 MHz, V_{DD} = 3.0 V Note 2	Square wave input		1.1	3.3	mA
				Resonator connection		1.4	3.6	
			f_{MX} = 10 MHz, V_{DD} = 3.0 V Notes 2, 3	Square wave input		0.55	2.1	mA
				Resonator connection		0.65	2.2	
			f_{MX} = 5 MHz, V_{DD} = 3.0 V Notes 2, 3	Square wave input		0.4	1.8	mA
				Resonator connection		0.45	1.8	
			f_{MX} = 5 MHz, V_{DD} = 2.0 V Notes 2, 3	Square wave input		0.26	1.3	mA
				Resonator connection		0.31	1.4	
			f_{IH} = 20 MHz Note 4	V_{DD} = 3.0 V		1.3	3.6	mA
			f_{IH} = 8 MHz Note 4	V_{DD} = 3.0 V		0.45	1.8	mA
			f_{IH} = 1 MHz, RMC = 5AH, OSMC = 02H Note 4	V_{DD} = 3.0 V		45	153	μA
			f_{SUB} = 32.768 kHz, RTCLPC = 1, FSEL = 0, SDIV = 1 Note 5	T_A = –40 to +50 °C, V_{DD} current		0.9	3.6	μA
				T_A = –40 to +50 °C, LV_{DD} current			15.0	
				T_A = –40 to +70 °C, V_{DD} current		0.9	6.0	μA
				T_A = –40 to +70 °C, LV_{DD} current			31.0	
				T_A = –40 to +85 °C, V_{DD} current		0.9	8.8	μA
				T_A = –40 to +85 °C, LV_{DD} current		2.5	47.0	

<R> **Notes 1.** Total current flowing into V_{DD}, AV_{DD}, LAV_{DD}, LV_{DD}, V_{LC0} to V_{LC2}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}, and excluding the current flowing into the watchdog timer, LVI circuit, A/D converter, LCD controller/driver, power calculation circuit, I/O port, and on-chip pull-up/pull-down resistors. The maximum values include the peripheral operation current. During HALT instruction execution by flash memory.

2. When internal high-speed oscillator and subsystem clock are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0 and FLPC, FSEL (bits 1, 0 of operation speed mode control register (OSMC)) = 0, 0.

4. When high-speed system clock and subsystem clock are stopped.

5. When internal high-speed oscillation, and high-speed system clock are stopped. When watchdog timer is <R> stopped. When real-time counter (RTC), real-time counter 2 (RTC2) are operating.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

DC Characteristics (8/9)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = LV_{DD} ≤ 3.6 V, V_{SS} = LV_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current	I_{DD3} Note 1	STOP mode	V_{DD} = LV_{DD} = 3.0 V Note 2	T_A = –40 to +50 °C	V_{DD} current		0.37	2.8	μA
					LV_{DD} current			5	
				T_A = –40 to +70 °C	V_{DD} current		0.37	5.2	μA
					LV_{DD} current			21	
				T_A = –40 to +85 °C	V_{DD} current		0.37	7.9	μA
					LV_{DD} current		0.2	37	
			V_{DD} = LV_{DD} = 3.0 V Note 3	T_A = –40 to +50 °C	V_{DD} current		0.37	2.8	μA
					LV_{DD} current			9	
				T_A = –40 to +70 °C	V_{DD} current		0.37	5.2	μA
					LV_{DD} current			25	
				T_A = –40 to +85 °C	V_{DD} current		0.37	7.9	μA
					LV_{DD} current		2.5	41	
RTC2 operating current	I_{RTC2} Notes 4, 5	f_{SUB} = 32.768 kHz			V_{DD} = 3.0 V		0.2	1	μA
					V_{DD} = 2.0 V		0.2	1	
RTC Operating current	I_{RTC} Note 6						0.4	2	μA
Watchdog timer operating current	I_{WDT} Notes 5, 7	f_{IL} = 30 kHz					0.31	0.35	μA
LVI operating current	I_{LVI} Note 8						9	18	μA
Successive approximation type A/D converter operating current	I_{ADC} Note 9	During conversion at maximum speed		Normal mode 1	AV_{DD0} = 3.0 V		0.7	1.4	mA

<R> **Notes 1.** Total current flowing into V_{DD}, AV_{DD}, LAV_{DD}, LV_{DD}, V_{LC0} to V_{LC2}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}, and excluding the current flowing into the watchdog timer, LVI circuit, A/D converter, LCD controller/driver, power calculation circuit, I/O port, and on-chip pull-up/pull-down resistors. The maximums values include the peripheral operation current and STOP leakage current.

2. When subsystem clock are stopped and watchdog timer, real-time counter (RTC) subsystem clock oscillation circuit are stopped.

<R> **3.** When subsystem clock is operating, but output from subsystem clock oscillation circuit is disabled (FXTOUTEN = 0). When watchdog timer, real-time counter (RTC) subsystem clock oscillation circuit are operating.

4. Current flowing only to the real-time counter 2 (V_{DD} pin) (excluding the operating current of the XT1 oscillator). The current value of the 78K0R/Lx3-M microcontrollers is the TYP. value, the sum of the TYP. values of either I_{DD1} or I_{DD2}, and I_{RTC2}, when the real-time counter 2 operates in an operation mode or HALT mode. The I_{DD1} and I_{DD2} MAX. values also include the real-time counter 2 operating current. When the real-time counter 2 operates during f_{CLK} = f_{SUBC}, the TYP. value of I_{DD2} includes the real-time counter 2 operating current.

5. When internal high-speed oscillator and high-speed system clock are stopped.

6. Current flowing only to real-time counter.

7. Current flowing only to the watchdog timer (V_{DD} pin) (including the operating current of the 30 kHz internal oscillator). The current value of the 78K0R/Lx3-M microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when f_{CLK} = f_{SUBC} or when the watchdog timer operates in STOP mode.

8. Current flowing only to the LVI circuit (V_{DD} pin). The current value of the 78K0R/Lx3-M microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVI circuit operates in the operation mode, HALT mode or STOP mode.

9. Current flowing only to the successive approximation type A/D converter (AV_{DD} pin). The current value of the 78K0R/Lx3-M microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the successive approximation type A/D converter operates in an operation mode or HALT mode.

<R> **DC Characteristics (9/9)**

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = LV_{DD} ≤ 3.6 V, V_{SS} = LV_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
LCD operating current	I_{LCD1} Notes 1, 2	External resistance division method	f_{LCD} = f_{SUB}, LCD panel not connected, LCD clock = 512 Hz		V_{DD} = 3.0 V		0.2	1.2	μA
	I_{LCD2} Note 1	Internal voltage boosting method	f_{LCD} = f_{SUB}, LCD panel not connected, LCD clock = 512 Hz	1/3 bias	V_{LCD} = 01H		1.39	4.7	μA
					V_{LCD} = 0FH		0.94	3.1	μA
	I_{LCD3} Note 1	Capacitor split method	f_{LCD} = f_{SUB}, LCD panel not connected, LCD clock = 512 Hz		V_{DD} = 3.0 V		0.36	1.7	μA
ΔΣ-type A/D converter operating current	I_{ADC2} Note 3						8.1	13.5	mA
Power calculation circuit operating current	I_{CAL} Notes 4, 5	Power calculation, power quality measurement, and DFC operating current.					3.7	4.5	mA

Notes 1. Current flowing only to the LCD controller/driver (V_{DD} pin). The current value of the 78K0R/Lx3-M microcontrollers is the sum of the LCD operating current (I_{LCD1}, I_{LCD2} or I_{LCD3}) to the supply current (I_{DD1}, or I_{DD2}) when the LCD controller/driver operates in an operation mode or HALT mode.

2. Not including the current that flows through the LCD divider resistor.

3. This includes only the current that flows through the ΔΣ-type A/D converter (AV_{DD} = 3.9 mA, LV_{DD} = 4.2 mA). When the ΔΣ-type A/D converter is operating in operation mode or HALT mode, the current value of the 78K0R/Lx3-M microcontrollers is obtained by adding I_{ADC2} to the supply current (I_{DD1} or I_{DD2}).

4. This includes only the current that flows through the power calculation circuit (LV_{DD} pin). When the power calculation circuit is operating, the current value of the 78K0R/Lx3-M microcontrollers is obtained by adding I_{CAL} to the supply current (I_{DD1}, I_{DD2}, or I_{DD3}).

5. If the typical value of V_{DD}, LAV_{DD}, and LV_{DD} is 3.3 V, this current flows to the LAV_{DD} and LV_{DD} pins.

AC Characteristics

(1) Basic operation (1/6)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = LV_{DD} ≤ 3.6 V, 1.8 V ≤ AV_{DD} ≤ V_{DD}, 1.8 V ≤ LAV_{DD} ≤ LV_{DD}, V_{SS} = LV_{SS} = LAV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	Normal power mode, FSEL = 1	2.7 V ≤ V_{DD} ≤ 3.6 V	0.05		8	μs
				1.8 V ≤ V_{DD} < 2.7 V	0.2		8	μs
			Normal power mode, FSEL = 0	2.7 V ≤ V_{DD} ≤ 3.6 V	0.1		8	μs
				1.8 V ≤ V_{DD} < 2.7 V	0.2		8	μs
			Low consumption power mode		1		8	μs
		Subsystem clock (f_{SUB}) operation	SDIV = 1		57.2	61	62.5	μs
			SDIV = 0		28.5	30.5	31.3	μs
		In the self programming mode	Normal power mode, FSEL = 1	2.7 V ≤ V_{DD} ≤ 3.6 V	0.05		1	μs
				1.8 V ≤ V_{DD} < 2.7 V	0.2		1	μs
			Low consumption power mode[Note]		0.88	1	1.15	μs
External main system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 3.6 V			2.0		20.0	MHz
		1.8 V ≤ V_{DD} < 2.7 V			2.0		5.0	MHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 3.6 V			24			ns
		1.8 V ≤ V_{DD} < 2.7 V			96			ns
TI02, TI04, TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}				2/f_{MCK} + 10			ns
TO02, TO05, TO07 output frequency	f_{TO}	2.7 V ≤ V_{DD} ≤ 3.6 V					10	MHz
		1.8 V ≤ V_{DD} < 2.7 V					5	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}				1			μs
RESET, LRESET low-level width	t_{RSL}				10			μs

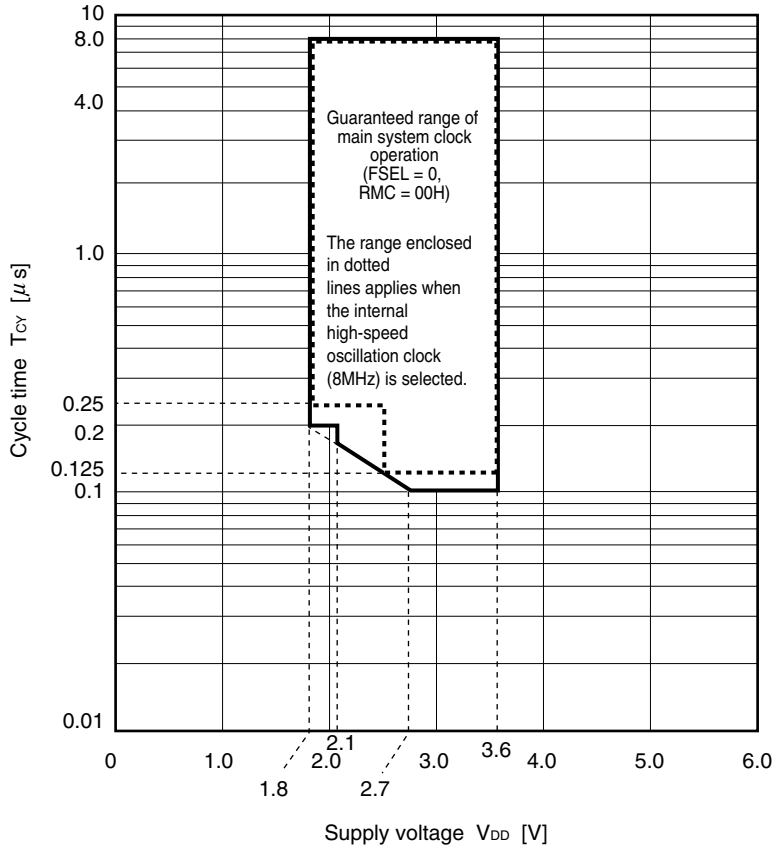
Note In low-power-consumption mode, use the regulator with f_{CLK} fixed to 1 MHz when executing self programming.

Remarks 1. f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the TMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 13)

2. For details on the normal power mode and low consumption power mode according to the regulator output voltage, refer to **CHAPTER 26 REGULATOR**.

(1) Basic operation (2/6)

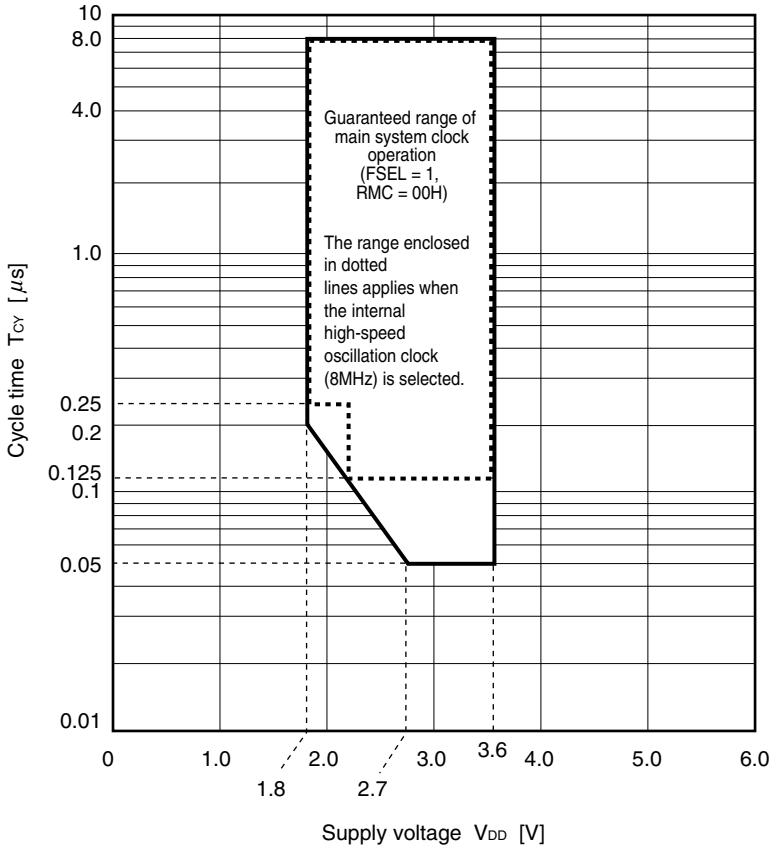
Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 00H)



Remark FSEL: Bit 0 of the operation speed mode control register (OSMC)

(1) Basic operation (3/6)

Minimum instruction execution time during main system clock operation (FSEL = 1, RMC = 00H)



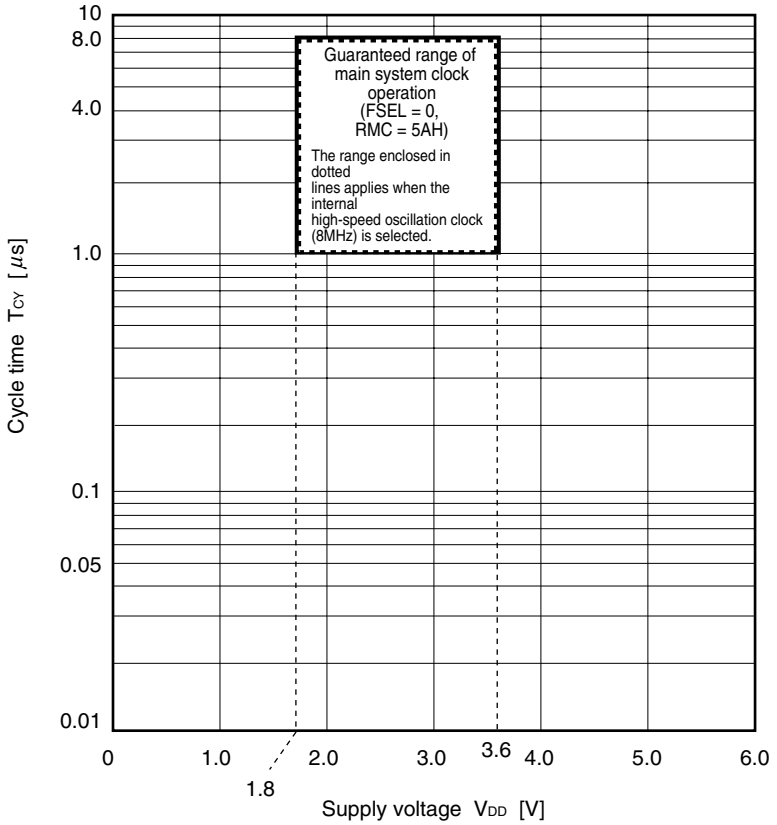
Caution When V_{DD} < 2.25 V and FSEL = 1, It is prohibited to release STOP mode during f_{EX} operation or f_{IH} operation (This must not be performed even if the frequency is divided. The STOP mode may be released during f_X operation.).

Remarks 1. FSEL: Bit 0 of the operation speed mode control register (OSMC)

2. f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency
f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency

(1) Basic operation (4/6)

Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 5AH)

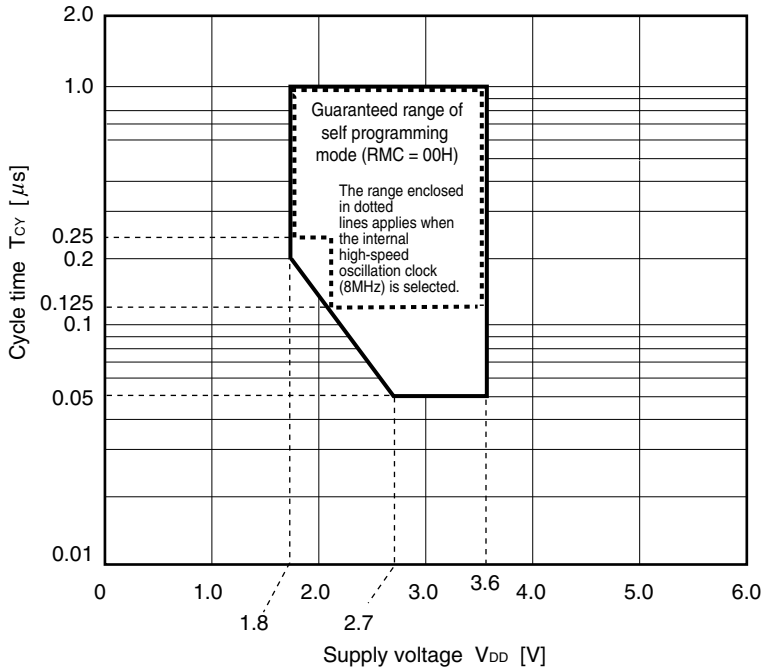


Remarks 1. FSEL: Bit 0 of the operation speed mode control register (OSMC)

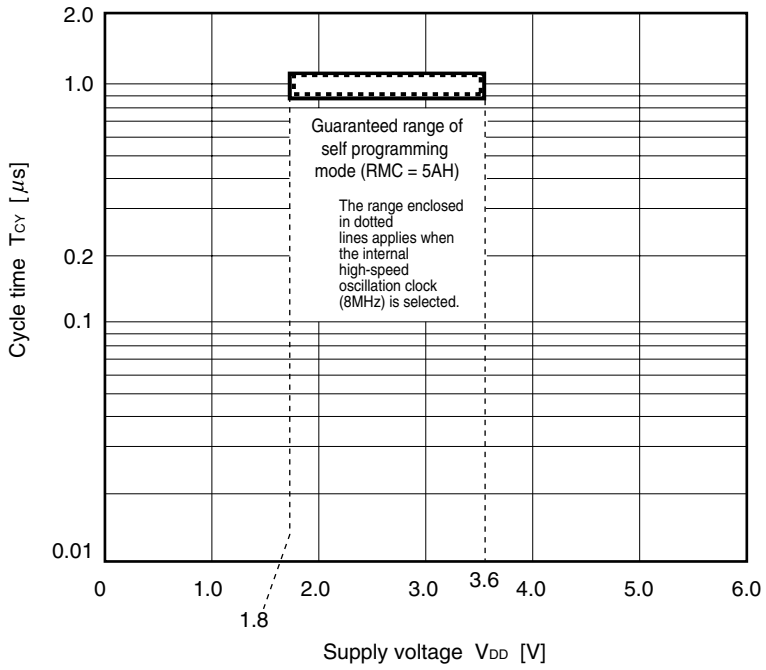
2. The entire voltage range is 1 MHz (MAX.) when RMC is set to 5AH.

(1) Basic operation (5/6)

Minimum instruction execution time during self programming mode (RMC = 00H)



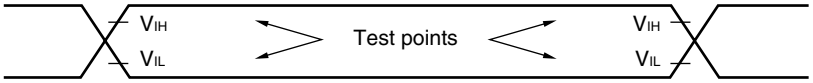
Minimum instruction execution time during self programming mode (RMC = 5AH)



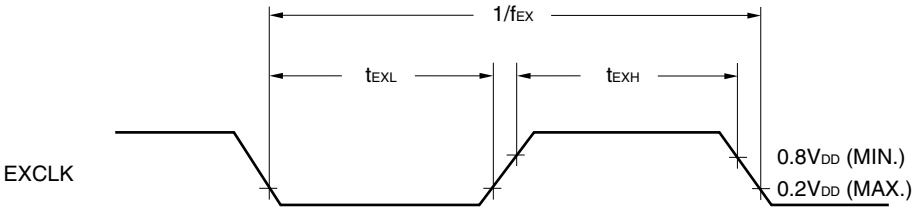
Remark The self programming function cannot be used when the CPU operates with the subsystem clock.

(1) Basic operation (6/6)

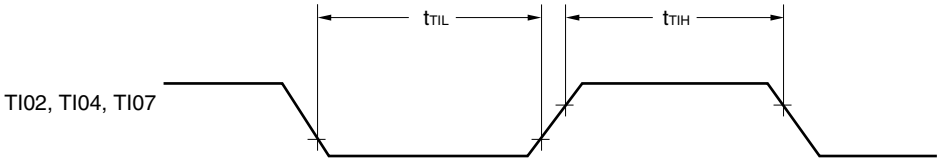
AC Timing Test Points



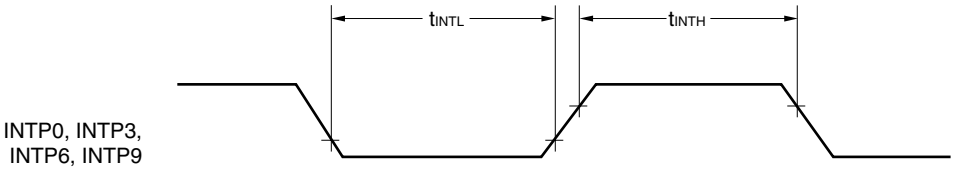
External Main System Clock Timing



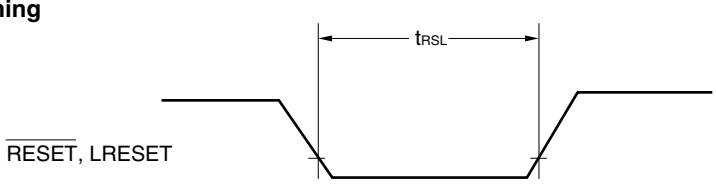
TI Timing



Interrupt Request Input Timing



$\overline{\text{RESET}}$ LRESET Input Timing



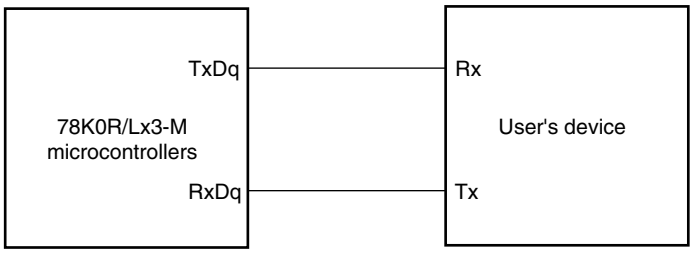
(2) Serial interface: Serial array unit (1/16)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

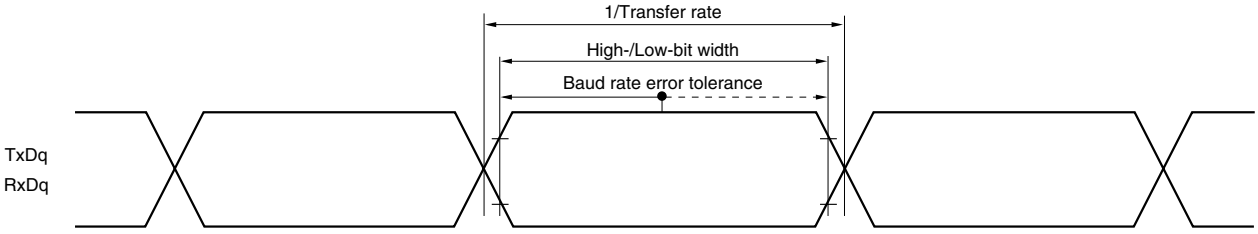
(a) During communication at same potential (UART mode) (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					f_{MCK}/6	bps
		f_{CLK} = 20 MHz, f_{MCK} = f_{CLK}			3.3	Mbps

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Caution Select the normal input buffer for RxDq and the normal output mode for TxDq by using the PIMg and POMx registers.

Remarks 1. q: UART number (q = 0, 2, 3), g: PIM number (g = 1), x: POM number (x = 1, 8)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2))

(2) Serial interface: Serial array unit (2/16)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

(b) During communication at same potential (CSI mode) (master mode, $\overline{SCKp}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY1}	2.7 V ≤ V_{DD} ≤ 3.6 V	300 Note 1			ns
		1.8 V ≤ V_{DD} < 2.7 V	600 Note 1			ns
$\overline{SCKp}$ high-/low-level width	t_{KH1}, t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 V	t_{KCY1}/2 – 35			ns
		1.8 V ≤ V_{DD} < 2.7 V	t_{KCY1}/2 – 80			ns
SIp setup time (to $\overline{SCKp}$↑) Note 2	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V	100			ns
		1.8 V ≤ V_{DD} < 2.7 V	190			ns
SIp hold time (from $\overline{SCKp}$↑) Note 3	t_{KSI1}		30			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 4	t_{KSO1}	C = 30 pF Note 5			40	ns

Notes 1. The value must also be 4/f_{CLK} or more.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

5. C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

Caution Select the normal input buffer for SIp and the normal output mode for SOp and $\overline{SCKp}$ by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 20), g: PIM number (g = 1), x: POM number (x = 1)

2. m: Unit number (m = 1), n: Channel number (n = 0)

(2) Serial interface: Serial array unit (3/16)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

(c) During communication at same potential (CSI mode) (slave mode, $\overline{SCKp}$... external clock input)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 3.6 V	16 MHz < f_{MCK}	8/f_{MCK}			ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}			ns
		1.8 V ≤ V_{DD} < 2.7 V	16 MHz < f_{MCK}	8/f_{MCK}			ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}			ns
$\overline{SCKp}$ high-/low-level width	t_{KH2}, t_{KL2}			t_{KCY2}/2			ns
SIp setup time (to $\overline{SCKp}$↑)[Note 1]	t_{SIK2}			80			ns
SIp hold time (from $\overline{SCKp}$↑)[Note 2]	t_{KSI2}			1/f_{MCK} + 50			ns
Delay time from $\overline{SCKp}$↓ to SOp output[Note 3]	t_{KSO2}	C = 30 pF[Note 4]	2.7 V ≤ V_{DD} = EV_{DD} ≤ 3.6 V			2/f_{MCK} + 57	ns
			1.8 V ≤ V_{DD} = EV_{DD} < 2.7 V			2/f_{MCK} + 125	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

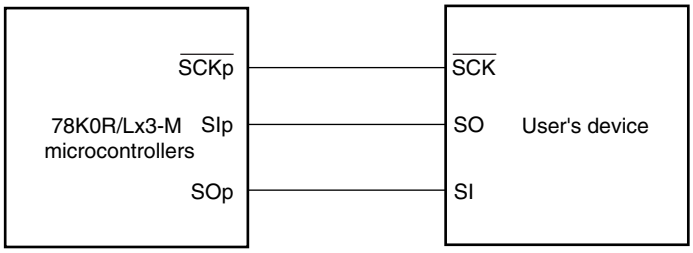
Caution Select the normal input buffer for SIp and $\overline{SCKp}$ and the normal output mode for SOp by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 20), g: PIM number (g = 1), x: POM number (x = 1)

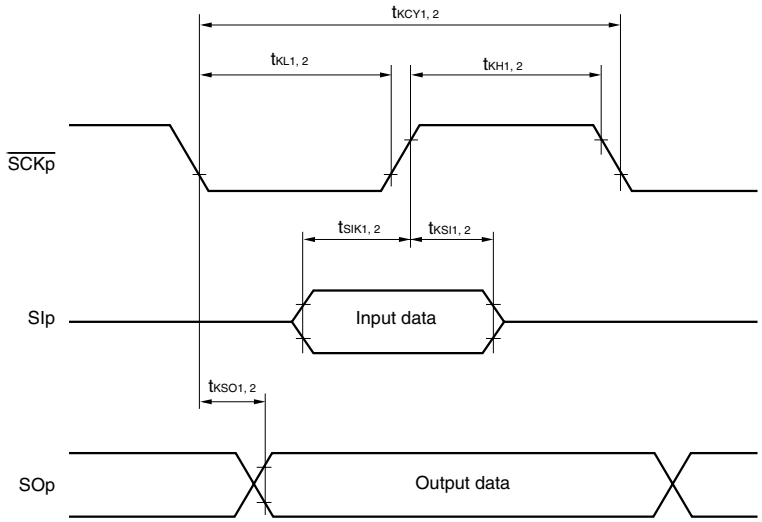
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 1), n: Channel number (n = 0))

(2) Serial interface: Serial array unit (4/16)

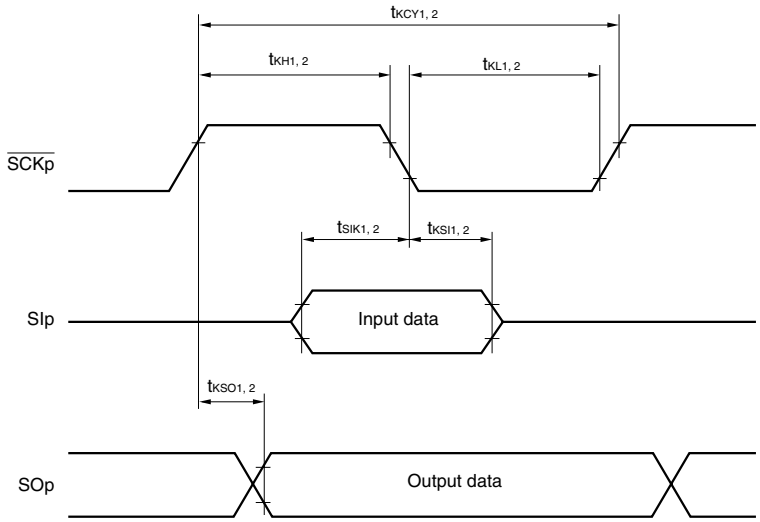
CSI mode connection diagram (during communication at same potential)



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 = 20)

2. m: Unit number (m = 1), n: Channel number (n = 0)

(2) Serial interface: Serial array unit (5/16)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

(d) During communication at same potential (simplified I^2C mode)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 3.6 V, R_b = 3 kΩ, C_b = 100 pF		400	kHz
		1.8 V ≤ V_{DD} ≤ 3.6 V, R_b = 5 kΩ, C_b = 100 pF		300	kHz
Hold time when SCLr = "L"	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V, R_b = 3 kΩ, C_b = 100 pF	1200		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, R_b = 5 kΩ, C_b = 100 pF	1500		ns
Hold time when SCLr = "H"	t_{HIGH}	2.7V ≤ V_{DD} ≤ 3.6 V, R_b = 3 kΩ, C_b = 100 pF	1200		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, R_b = 5 kΩ, C_b = 100 pF	1500		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7V ≤ V_{DD} ≤ 3.6 V, R_b = 3 kΩ, C_b = 100 pF	1/f_{MCK} + 120		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, R_b = 5 kΩ, C_b = 100 pF	1/f_{MCK} + 230		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7V ≤ V_{DD} ≤ 3.6 V, R_b = 3 kΩ, C_b = 100 pF	0	660	ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, R_b = 5 kΩ, C_b = 100 pF	0	710	ns

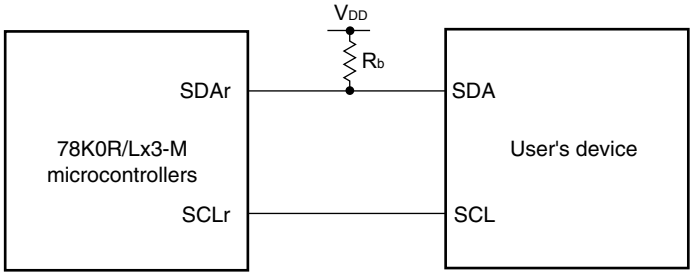
Caution Select the normal input buffer and the N-ch open drain output (V_{DD} tolerance) mode for SDAr and the normal output mode for SCLr by using the PIMg and POMx registers.

Remarks 1. R_b[Ω]:Communication line (SDAr) pull-up resistance,
C_b[F]: Communication line (SCLr, SDAr) load capacitance

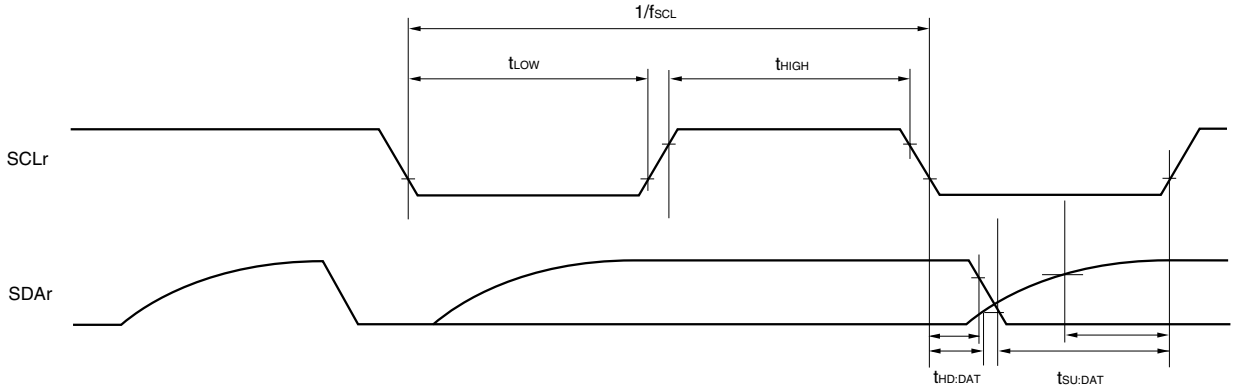
2. r: IIC number (r = 20), g: PIM number (g = 1), x: POM number (x = 1)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 1), n: Channel number (n = 0), mn = 10)

Simplified I^2C mode connection diagram (during communication at same potential)



Simplified I^2C mode serial transfer timing (during communication at same potential)



Remarks 1. R_b[Ω]:Communication line (SDAr) pull-up resistance,
C_b[F]: Communication line (SCLr, SDAr) load capacitance

2. r: IIC number (r = 20)

(2) Serial interface: Serial array unit (6/16)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

(e) Communication at different potential (2.5 V) (UART mode) (dedicated baud rate generator output) (1/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Transfer rate		reception	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V				f_{MCK}/6	bps
				f_{CLK} = 20 MHz, f_{MCK} = f_{CLK}			3.3	Mbps

Caution Select the TTL input buffer for RxDq and the N-ch open drain output (V_{DD} tolerance) mode for TxDq by using the PIMg and POMx registers.

Remarks 1. q: UART number (q = 0, 2, 3), g: PIM number (g = 1), x: POM number (x = 1, 8)

2. V_b[V]: Communication line voltage

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3))

4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.
2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (7/16)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

(e) Communication at different potential (2.5 V) (UART mode) (dedicated baud rate generator output) (2/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Transfer rate		transmission	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V				Note 1	bps
				f_{CLK} = 19.2 MHz, f_{MCK} = f_{CLK}, C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V			1.2 Note 2	Mbps

Notes 1. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} ≤ 3.6 V and 2.3 V ≤ V_b < 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1-\frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1-\frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

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.

Caution Select the TTL input buffer for RxDq and the N-ch open drain output (V_{DD} tolerance) mode for TxDq by using the PIMg and POMx registers.

Remarks 1. R_b[Ω]:Communication line (TxDq) pull-up resistance, C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

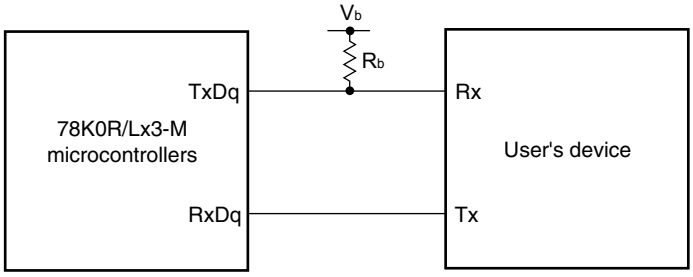
2. q: UART number (q = 0, 2, 3) , g: PIM number (g = 1), x: POM number (x = 1, 8)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register.
m: Unit number (m = 0, 1), n: Channel number (n = 0, 2))

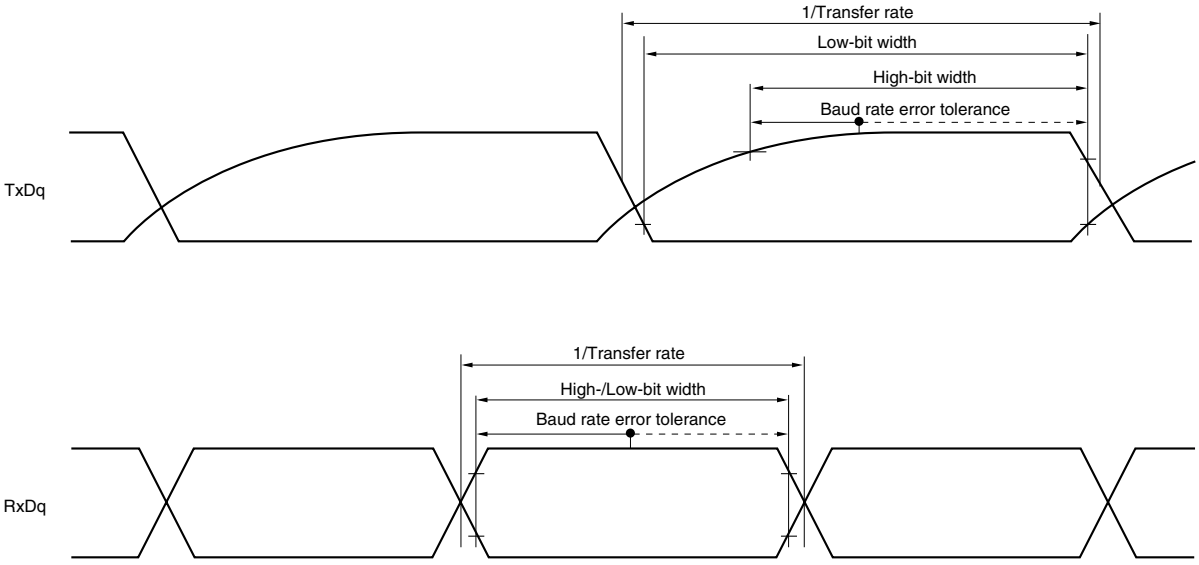
4. V_{OH} and V_{OL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.
2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V: V_{OH} = 2.0 V, V_{OL} = 0.5 V

(2) Serial interface: Serial array unit (8/16)

UART mode connection diagram (communication at different potential)



UART mode bit width (communication at different potential) (reference)



Caution Select the TTL input buffer for RxDq and the N-ch open drain output (V_{DD} tolerance) mode for TxDq by using the PIMg and POMx registers.

Remarks 1. $R_b[\Omega]$:Communication line (TxDq) pull-up resistance, $V_b[V]$: Communication line voltage

2. q: UART number (q = 0, 2, 3) , g: PIM number (g = 1), x: POM number (x = 1, 8)

(2) Serial interface: Serial array unit (9/16)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

(f) Communication at different potential (2.5 V) (CSI mode) (master mode, $\overline{SCKp}$... internal clock output) (1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	800 Note 1			ns
$\overline{SCKp}$ high-level width	t_{KH1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 170			ns
$\overline{SCKp}$ low-level width	t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 35			ns
SIp setup time (to $\overline{SCKp}$↑) Note 2	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	275			ns
SIp hold time (from $\overline{SCKp}$↑) Note 2	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 2	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			215	ns

Notes 1. The value must also be 4/f_{CLK} or more.

2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1.

Caution Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and $\overline{SCKp}$ by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 20), g: PIM number (g = 1), x: POM number (x = 1)

2. m: Unit number (m = 1), n: Channel number (n = 0)

3. R_b[Ω]:Communication line ($\overline{SCKp}$, SOp) pull-up resistance,
C_b[F]: Communication line (SIp, SOp, $\overline{SCKp}$) load capacitance, V_b[V]: Communication line voltage

4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.
2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (10/16)

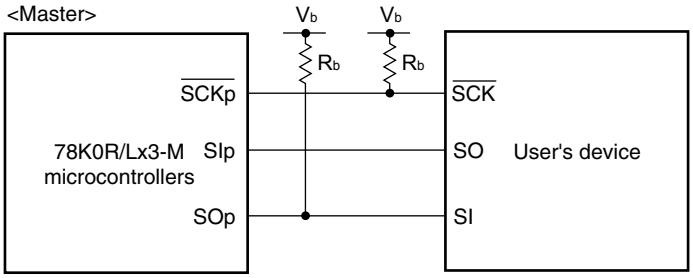
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

(f) Communication at different potential (2.5 V) (CSI mode) (master mode, $\overline{SCKp}$... internal clock output) (2/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SIp setup time (to $\overline{SCKp}$↓)Note	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	100			ns
SIp hold time (from $\overline{SCKp}$↓)Note	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{SCKp}$↑ to SOp outputNote	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			40	ns

Note When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

CSI mode connection diagram (communication at different potential)



Caution Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and $\overline{SCKp}$ by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 20), g: PIM number (g = 1), x: POM number (x = 1)

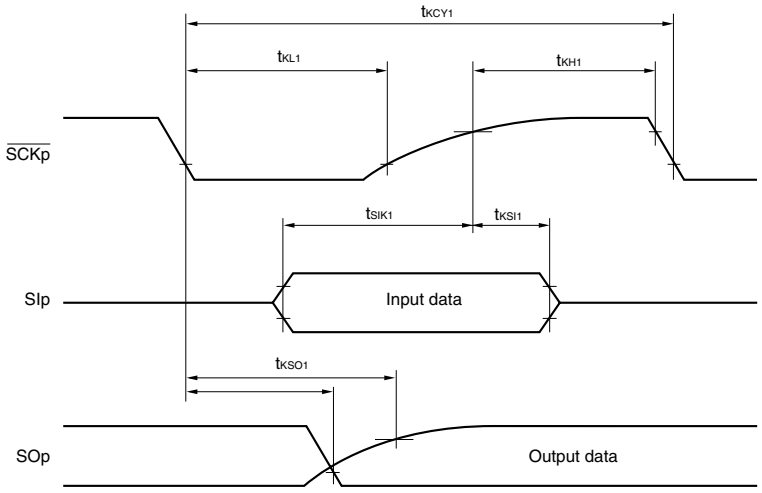
2. m: Unit number (m = 1), n: Channel number (n = 0)

3. R_b[Ω]:Communication line ($\overline{SCKp}$, SOp) pull-up resistance,
C_b[F]: Communication line (SIp, SOp, $\overline{SCKp}$) load capacitance, V_b[V]: Communication line voltage

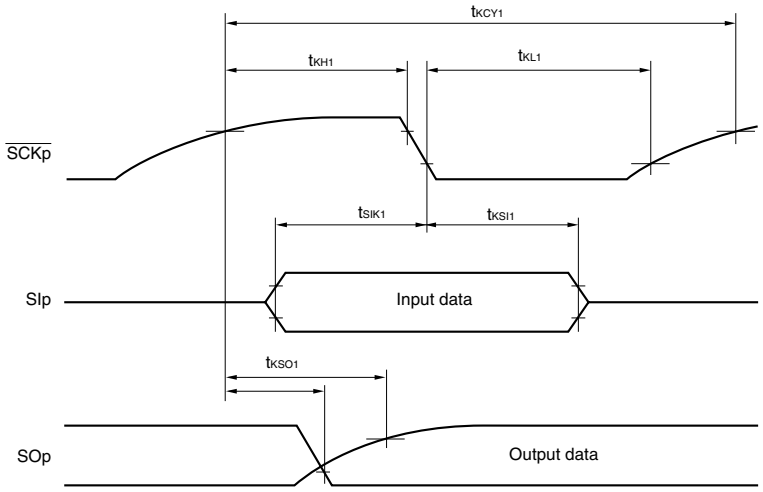
4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.
2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (11/16)

CSI mode serial transfer timing (communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Caution Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and $\overline{SCKp}$ by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 20), g: PIM number (g = 1), x: POM number (x = 1)

2. m: Unit number (m = 1), n: Channel number (n = 0)

(2) Serial interface: Serial array unit (12/16)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

(g) Communication at different potential (2.5 V) (CSI mode) (slave mode, $\overline{SCKp}$... external clock input)

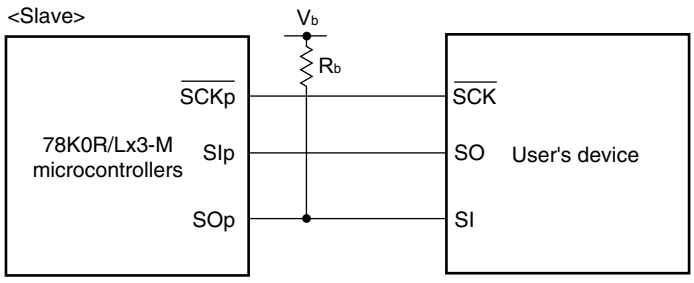
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V	18.5 MHz < f_{MCK}	16/f_{MCK}			ns
			14.8 MHz < f_{MCK} ≤ 18.5 MHz	14/f_{MCK}			ns
			11.1 MHz < f_{MCK} ≤ 14.8 MHz	12/f_{MCK}			ns
			7.4 MHz < f_{MCK} ≤ 11.1 MHz	10/f_{MCK}			ns
			3.7 MHz < f_{MCK} ≤ 7.4 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 3.7 MHz	6/f_{MCK}			ns
$\overline{SCKp}$ high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 – 35			ns
SIp setup time (to $\overline{SCKp}$↑)[Note 1]	t_{SIK2}			90			ns
SIp hold time (from $\overline{SCKp}$↑)[Note 2]	t_{KSI2}			1/f_{MCK} + 50			ns
Delay time from $\overline{SCKp}$↓ to SOp output[Note 3]	t_{KSO2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ				2/f_{MCK} + 230	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

CSI mode connection diagram (communication at different potential)



Caution Select the TTL input buffer for SIp and $\overline{SCKp}$ and the N-ch open drain output (V_{DD} tolerance) mode for SOp by using the PIMg and POMx registers.

(**Remarks** are given on the next page.)

(2) Serial interface: Serial array unit (13/16)

Remarks 1. p: CSI number (p = 20), g: PIM number (g = 1), x: POM number (x = 1)

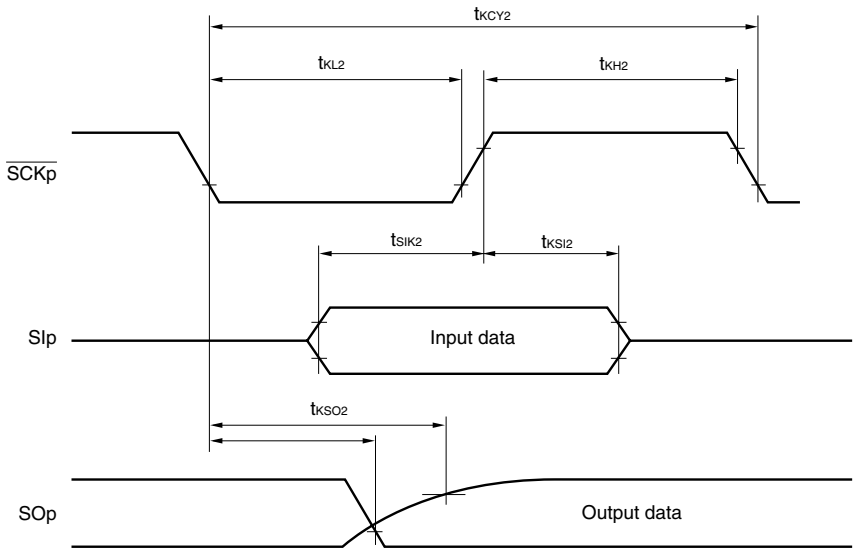
2. R_b[Ω]:Communication line (SOp) pull-up resistance,
C_b[F]: Communication line (SOp, $\overline{SCKp}$) load capacitance, V_b[V]: Communication line voltage

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 1), n: Channel number (n = 0))

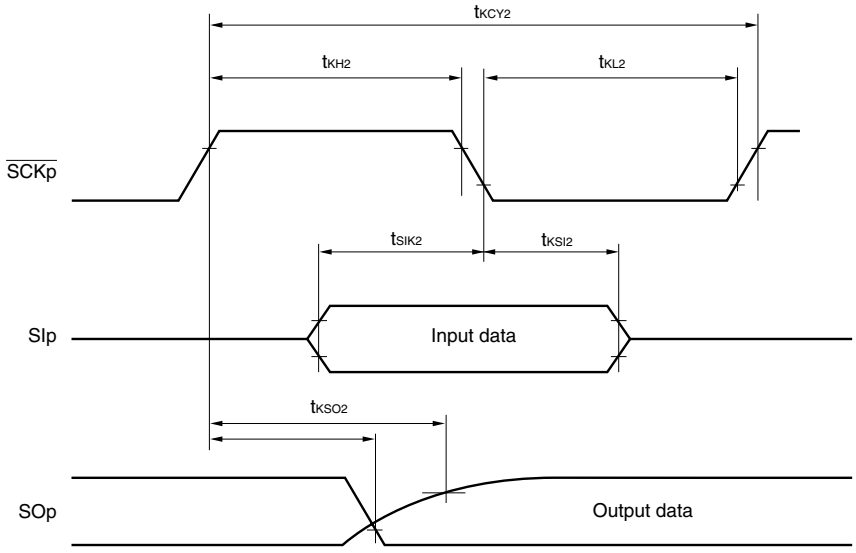
4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.
2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (14/16)

CSI mode serial transfer timing (communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Caution Select the TTL input buffer for SIp and $\overline{\text{SCKp}}$ and the N-ch open drain output (V_{DD} tolerance) mode for SOp by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 20), g: PIM number (g = 1), x: POM number (x = 1)

2. m: Unit number (m = 1), n: Channel number (n = 0)

(2) Serial interface: Serial array unit (15/16)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

(h) Communication at different potential (2.5 V) (simplified I^2C mode)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, R_b = 2.7 kΩ, C_b = 100 Pf		400	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, R_b = 2.7 kΩ, C_b = 100 pF,	1275		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, R_b = 2.7 kΩ, C_b = 100 pF	655		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, R_b = 2.7 kΩ, C_b = 100 pF	1/f_{MCK} + 190		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, R_b = 2.7 kΩ, C_b = 100 pF	0	660	ns

Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for SDAr and the N-ch open drain output (V_{DD} tolerance) mode for SCLr by using the PIMg and POMx registers.

Remarks 1. R_b[Ω]:Communication line (SDAr, SCLr) pull-up resistance,
C_b[F]: Communication line (SDAr, SCLr) load capacitance, V_b[V]: Communication line voltage

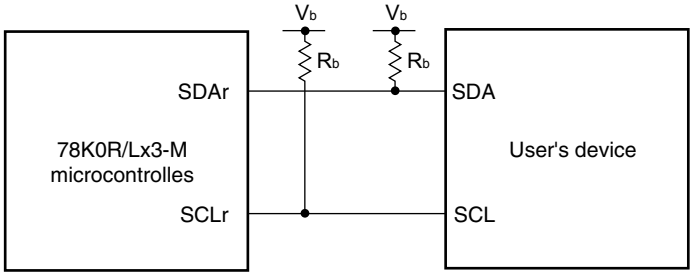
2. r: IIC number (r = 20), g: PIM number (g = 1), x: POM number (x = 1)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 1),
n: Channel number (n = 0), mn = 10)

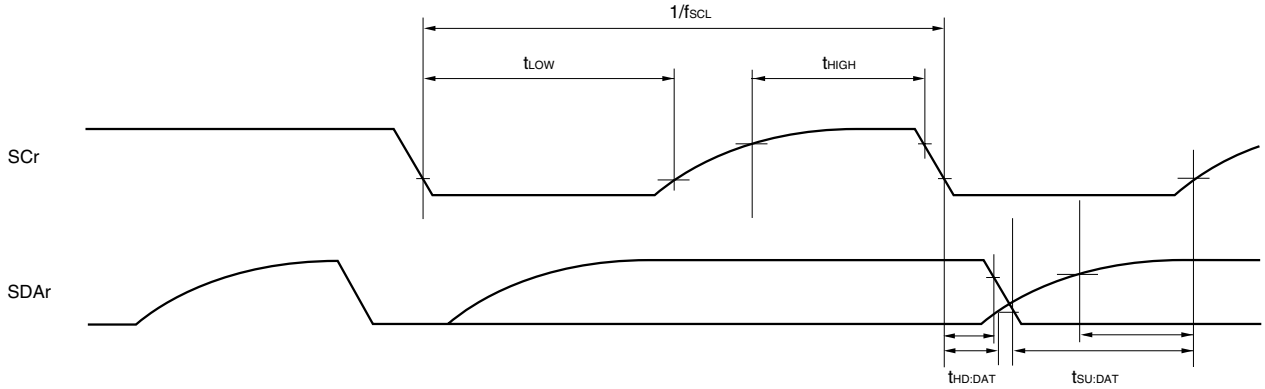
4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in simplified I^2C mode.
2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (16/16)

Simplified I^2C mode connection diagram (communication at different potential)



Simplified I^2C mode serial transfer timing (communication at different potential)



Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for SDAr and the N-ch open drain output (V_{DD} tolerance) mode for SCLr by using the PIMg and POMx registers.

Remarks 1. R_b[Ω]:Communication line (SDAr, SCLr) pull-up resistance, V_b[V]: Communication line voltage

2. r: IIC number (r = 20), g: PIM number (g = 1), x: POM number (x = 1)

(3) Serial interface: IICA

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

(a) IICA

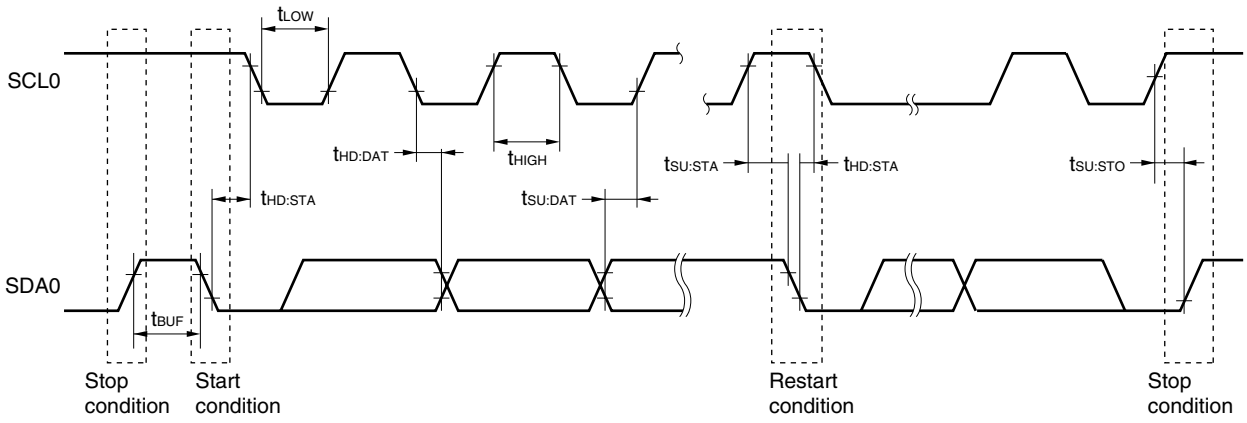
Parameter	Symbol	Conditions	Standard Mode		High-Speed Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCL0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥3.5 MHz, Standard mode: f_{CLK} ≥1 MHz	0	100	0	400	kHz
Setup time of restart condition[Note 1]	$t_{SU:STA}$		4.7		0.6		μs
Hold time	$t_{HD:STA}$		4.0		0.6		μs
Hold time when SCL0 = “L”	t_{LOW}		4.7		1.3		μs
Hold time when SCL0 = “H”	t_{HIGH}		4.0		0.6		μs
Data setup time (reception)	$t_{SU:DAT}$		250		100		ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0	3.45	0	0.9	μs
Setup time of stop condition	$t_{SU:STO}$		4.0		0.6		μs
Bus-free time	t_{BUF}		4.7		1.3		μs

Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a wait state is inserted in the $\overline{ACK}$ (acknowledge) timing.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

IICA serial transfer timing



(4) Serial interface: On-chip debug (UART)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

(a) On-chip debug (UART)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate			$f_{CLK}/2^{12}$		$f_{CLK}/6$	bps
		Flash memory programming mode (f_{CLK} = 20 MHz, 2.7 V ≤ V_{DD}, C_b = 50 pF)			3.33	Mbps
TOOL1 output frequency	f_{TOOL1}	2.7 V ≤ V_{DD} ≤ 3.6 V			10	MHz
		1.8 V ≤ V_{DD} < 2.7 V			2.5	MHz

Analog Characteristics

(1) 10-bit successive approximation type A/D converter

(T_A = –40 to +85°C , 1.8 V ≤ AV_{REF} ≤ AV_{DD} ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES		10	10	10	bit
Overall error[Note]	AINL				±0.4	%FSR
Conversion time	t_{CONV}	Normal mode 1	5		50	μs
Zero-scale error[Note]	E_{ZS}				±0.4	%FSR
Full-scale error[Note]	E_{FS}				±0.4	%FSR
Integral non-linearity error[Note]	ILE				±2.5	LSB
Differential non-linearity error[Note]	DLE				±1.5	LSB
Analog input voltage	V_{AIN}		AV_{SS}		AV_{REF}	V

Note Excludes quantization error (±1/2 LSB).

(2) 24-bit ΔΣ-type A/D Converter Characteristics

(T_A = –40 to +85°C, 3.0 V ≤ LAV_{DD} ≤ LV_{DD} ≤ 3.6 V, LV_{SS} = LAV_{SS} = 0 V)

(a) Recommended operation condition

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Supply voltage	LAV_{DD}		3.0	3.3	3.6	V
	LV_{DD}		3.0	3.3	3.6	V
Clock frequency	CLK_{ORG}	For decimation filter		10.0		MHz
Analog input voltage	$V_{ANI (V)}$	Voltage channel	–0.375		0.375	V
	$V_{ANI (I)}$	Current channel	–0.1875		0.1875	V
Operating temperature	T_{OPR}		–40	25	85	°C

(b) Reference

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
External reference potential (input)	AV_{REFIO1}		1.20		1.25	V
Internal reference potential (output)	AV_{REFIO2}		1.165	1.226	1.287	V
Internal reference potential temperature coefficient	dREF/dT	**Note**		100		ppm/°C

Note Temperature coefficient of –40 to +25°C and +25 to +85°C

(c) Analog input

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input signal DC level	ainDC	Voltage channel	–20	0	20	mV
		Current channel, gain 1 time[Note]	–20	0	20	mV
		Current channel, gain 2 times	–10	0	10	mV
		Current channel, gain 16 times	–1.25	0	1.25	mV
Input signal range	ainRANGE	Voltage channel	–0.375	0	0.375	V
		Current channel, gain 1 time[Note]	–0.375	0	0.375	V
		Current channel, gain 2 times	–0.1875	0	0.1875	V
		Current channel, gain 16 times	–23.4375	0	23.4375	mV
Input gain	ainGAIN	Voltage channel	0.93	1	1.07	Times
		Current channel, gain 1 time[Note]	0.93	1	1.07	Times
		Current channel, gain 2 times	1.86	2	2.14	Times
		Current channel, gain 16 times	14.88	16	17.12	Times
Input impedance	ainRIN	Voltage channel	100	131.25		kΩ
		Current channel, gain 1 time[Note]	100	131.25		kΩ
		Current channel, gain 2 times	60	78.75		kΩ
		Current channel, gain 16 times	60	78.75		kΩ

Note When channel 2 is used as current channel (See **CHAPTER 11 24-BIT ΔΣ-TYPE A/D CONVERTER**).

Caution The programmable amplifier gain specification of channel 1 and 3 must be the same gain setting. Channel 2 is fixed with gain 1 time.

(d) A/D converter part

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
System clock	CLK_{ORG}				10		MHz
ΔΣ-type operating clock	CLK_{OSX3}	CLK_{OSX6}/2			1.667		MHz
Oversampling frequency	f_{OS}				555.6		kHz
Sampling frequency	f_S	f_{OS}/128			4.34		kHz
Output data rate	T_{DATA}	At 1/f_S DF output			230.4		μs
Data width	RES				24		bit
S/N	SNR	0 dB@60 Hz When inputting a single sine wave	Voltage channel	70	76		dB
			Current channel, gain 1 time[Note]	70	76		dB
			Current channel, gain 2 times	70	76		dB
			Current channel, gain 16 times	62	69		dB
THD	THD	0 dB@60 Hz When inputting a single sine wave	Voltage channel		–80	–72	dB
			Current channel, gain 1 time[Note]		–80	–72	dB
			Current channel, gain 2 times		–80	–72	dB
			Current channel, gain 16 times		–80	–72	dB
Inter-channel isolation	XT		Voltage channel	80			dB
			Current channel, gain 1 time[Note]	80			dB
			Current channel, gain 2 times	80			dB
			Current channel, gain 16 times	72			dB
Operating current	I_{AVDD}				3.9	7	mA

Note When channel 2 is used as current channel (See **CHAPTER 11 24-BIT ΔΣ-TYPE A/D CONVERTER**).

Caution Initial data must be treated as follows:

- **Power off → Power on → Start the conversion → Discard the initial data (INTAD2 in 1200 times) → Obtain data (INTAD2 of 1201st time)**
- **Power on → Stop the conversion → Start the conversion → Discard the initial data (INTAD2 in 10 times) → Obtain data (INTAD2 of 11th time)**

(e) Digital filter part

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Passband (low)	fchpf	–3 dB		5.4		Hz
In-band ripple 1	rp1	45 Hz to 55 Hz@50 Hz 54 Hz to 66 Hz@60 Hz	–0.03		0.03	dB
In-band ripple 2	rp2	45 Hz to 275 Hz@50 Hz 54 Hz to 330 Hz@60 Hz	–0.1		0.1	dB
In-band ripple 3	rp3	45 Hz to 1100 Hz@50 Hz 54 Hz to 1320 Hz@60 Hz	–0.1		0.1	dB
Stopband (high)	fatt	–80 dB		3020		Hz
Out-of-band attenuation	ATT		–80			dB

Caution All current channels must have the same gain setting (It is not possible to specify different gain settings for each channel).

Power Calculation Characteristics

(T_A = –40 to +85°C, 3.0 V ≤ LV_{DD} ≤ 3.6 V, LV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Phase error between channels PF = 0.8 capacitive		Phase lead: 37°		±0.05		°
Phase error between channels PF = 0.5 inductive		Phase lag: 60°		±0.05		°
Active energy measurement error		Dynamic range 1000: 1 @ 25°C PF = 1		±0.1		%
Reactive energy measurement error		Dynamic range 1000: 1 @ 25°C PF = 0		±0.5		%
Vrms measurement error		Dynamic range 100: 1 @ 25°C		±0.5		%
Irms measurement error		Dynamic range 500: 1 @ 25°C		±0.5		%

LCD Characteristics (1/3)

(1) Resistance division method

(a) Static display mode (T_A = –40 to +85°C, V_{LCD} (MIN.) ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{LCD}		2.0		V_{DD}	V
LCD output resistor[Note] (Common)	R_{ODC}	I_O = ±5 μA			40	kΩ
LCD output resistor[Note] (Segment)	R_{OCS}	I_O = ±1 μA			200	kΩ

(b) 1/2 bias method (T_A = –40 to +85°C, V_{LCD} (MIN.) ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{LCD}		2.7		V_{DD}	V
LCD output resistor[Note] (Common)	R_{ODC}	I_O = ±5 μA			40	kΩ
LCD output resistor[Note] (Segment)	R_{OCS}	I_O = ±1 μA			200	kΩ

(c) 1/3 bias method (T_A = –40 to +85°C, V_{LCD} (MIN.) ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{LCD}		2.5		V_{DD}	V
LCD output resistor[Note] (Common)	R_{ODC}	I_O = ±5 μA			40	kΩ
LCD output resistor[Note] (Segment)	R_{OCS}	I_O = ±1 μA			200	kΩ

Note The output resistor is a resistor connected between one of the V_{LC0}, V_{LC1}, V_{LC2}, and V_{SS} pins, and either of the SEG and COM pins.

LCD Characteristics (2/3)

(2) Internal voltage boosting method

• 1/3 bias method (T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD output voltage variation range	V_{LCD2}	C1 to C4[Note 1] = 0.47 μF[Note 2]	VLCD = 00H	1.67	1.75	1.83	V
			VLCD = 01H	1.62	1.70	1.78	V
			VLCD = 02H	1.57	1.65	1.73	V
			VLCD = 03H	1.52	1.60	1.68	V
			VLCD = 04H	1.47	1.55	1.63	V
			VLCD = 05H	1.42	1.50	1.58	V
			VLCD = 06H	1.37	1.45	1.53	V
			VLCD = 07H	1.32	1.40	1.48	V
			VLCD = 08H	1.27	1.35	1.43	V
			VLCD = 09H	1.22	1.30	1.375	V
			VLCD = 0AH	1.17	1.25	1.33	V
			VLCD = 0BH	1.12	1.20	1.28	V
			VLCD = 0CH	1.07	1.15	1.23	V
			VLCD = 0DH	1.02	1.10	1.18	V
			VLCD = 0EH	0.97	1.05	1.13	V
			VLCD = 0FH	0.92	1.00	1.08	V
			VLCD = 10H	0.87	0.95	1.03	V
			VLCD = 11H	0.82	0.90	0.98	V
			VLCD = 12H	0.77	0.85	0.93	V
			VLCD = 13H	0.72	0.80	0.88	V
Doubler output voltage	V_{LCD1}	C1 to C4[Note 1] = 0.47 μF		2 V_{LCD2} –0.1	2 V_{LCD2}	2 V_{LCD2}	V
Tripler output voltage	V_{LCD0}	C1 to C4[Note 1] = 0.47 μF		3 V_{LCD2} –0.15	3 V_{LCD2}	3 V_{LCD2}	V
Reference voltage setup time[Note 2]	$t_{VAWAIT2}$			2			ms
Voltage boost wait time[Note 3]	$t_{VAWAIT1}$			500			ms
		V_{DD} > V_{LC0}		5			s
LCD output resistor[Note 4] (Common)	R_{ODC}	Io = ±5 μA				40	kΩ
LCD output resistor[Note 4] (Segment)	R_{OCS}	Io = ±1 μA				200	kΩ

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_{LC0} and GND
C3: A capacitor connected between V_{LC1} and GND
C4: A capacitor connected between V_{LC2} and GND
C1 = C2 = C3 = C4 = 0.47 pF±30 %

2. This is the required wait time from when the reference voltage is specified by using the LVCD register (or the register is reset to use the default value of the reference voltage) until voltage boosting is started (VLCON = 1).

3. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

4. The output resistor is a resistor connected between one of the V_{LC0}, V_{LC1}, V_{LC2} and V_{SS} pins, and either of the SEG and COM pins.

LCD Characteristics (3/3)

(3) Capacitor split method

- **1/3 bias method (T_A = –40 to +85°C, 2.2 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{LC0} voltage	V_{LC0}	C1 to C4 = 0.47 μF[Note 3]		V_{DD}		V
V_{LC1} voltage	V_{LC1}	C1 to C4 = 0.47 μF[Note 3]	2/3 V_{LC0} –0.1	2/3 V_{LC0}	2/3 V_{LC0} + 0.1	V
V_{LC2} voltage	V_{LC2}	C1 to C4 = 0.47 μF[Note 3]	1/3 V_{LC0} –0.1	1/3 V_{LC0}	1/3 V_{LC0} + 0.1	V
Capacitor split wait time[Note 1]	t_{VAWAIT}		100			ms
LCD output resistor[Note 2] (Common)	R_{ODC}	Io = ±5 μA			40	kΩ
LCD output resistor[Note 2] (Segment)	R_{OCS}	Io = ±1 μA			200	kΩ

Notes 1. This is the wait time from when voltage bucking is started (VLCON = 1) until display is enabled (LCDON = 1).

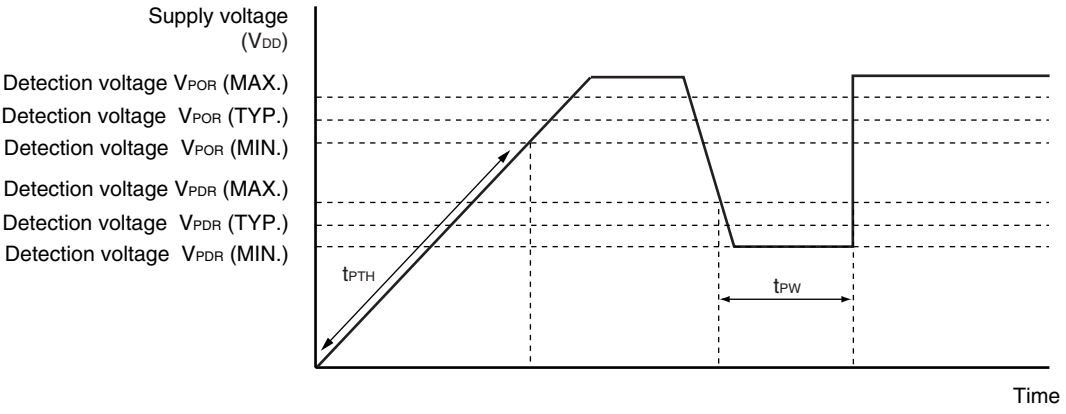
2. The output resistor is a resistor connected between one of the V_{LC0}, V_{LC1}, V_{LC2} and V_{SS} pins, and either of the SEG and COM pins.

3. 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_{LC0} and GND
C3: A capacitor connected between V_{LC1} and GND
C4: A capacitor connected between V_{LC2} and GND
C1 = C2 = C3 = C4 = 0.47 pF±30 %

POC Circuit Characteristics (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}		1.52	1.61	1.70	V
	V_{PDR}		1.5	1.59	1.68	V
Power supply voltage rise inclination	t_{PTH}	Change inclination of V_{DD}: 0 V → V_{POR}	0.5			V/ms
Minimum pulse width	t_{PW}	When the voltage drops	200			μs
Detection delay time					200	μs

POC Circuit Timing



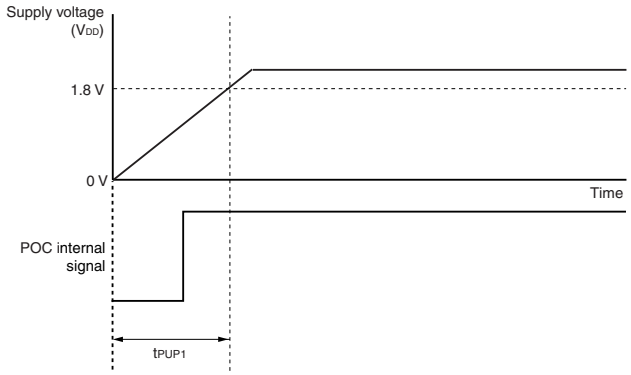
Supply Voltage Rise Time (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 1.8 V (V_{DD} (MIN.))[Note] (V_{DD}: 0 V → 1.8 V)	t_{PUP1}	LVI default start function stopped is set (LVIOFF (Option Byte) = 1), when $\overline{RESET}$ input is not used			3.6	ms
Maximum time to rise to 1.8 V (V_{DD} (MIN.))[Note] (releasing $\overline{RESET}$ input → V_{DD}: 1.8 V)	t_{PUP2}	LVI default start function stopped is set (LVIOFF (Option Byte) = 1), when $\overline{RESET}$ input is used			1.88	ms

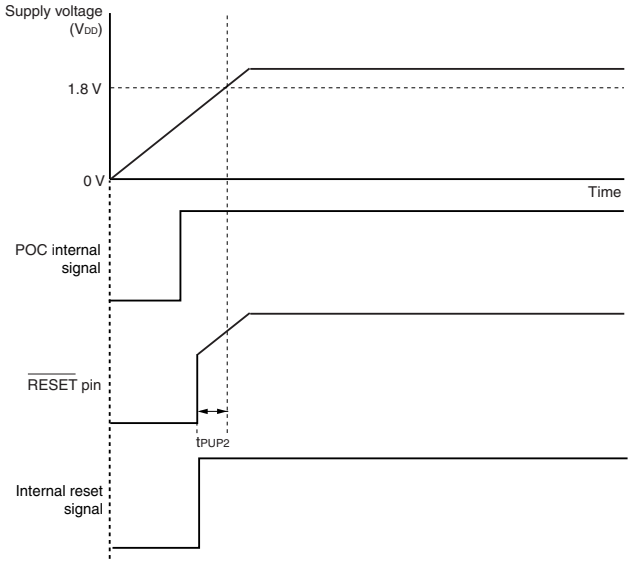
Note Make sure to raise the power supply in a shorter time than this.

Supply Voltage Rise Time Timing

- When $\overline{RESET}$ pin input is not used



- When $\overline{RESET}$ pin input is used (when external reset is released by the $\overline{RESET}$ pin, after POC has been released)



LVI Circuit Characteristics (T_A = –40 to +85°C, $V_{PDR} \le V_{DD} \le 3.6$ V, V_{SS} = 0 V)

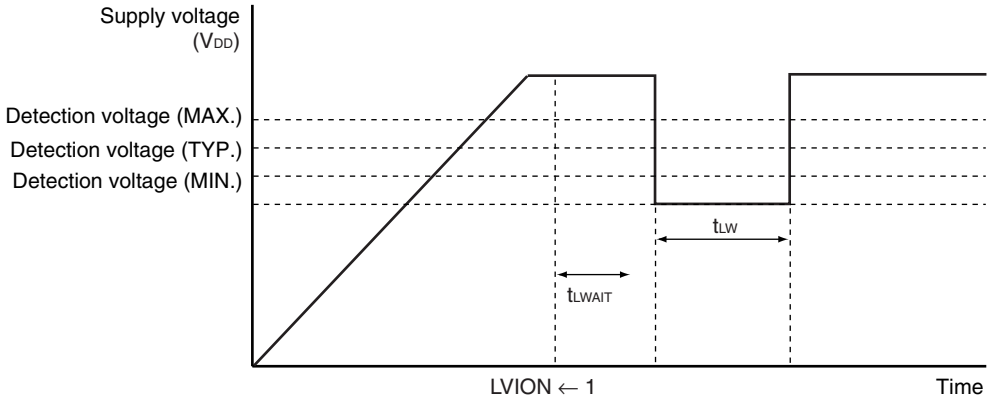
Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		3.35	3.45	3.55	V
		V_{LVI1}		3.20	3.30	3.40	V
		V_{LVI2}		3.05	3.15	3.25	V
		V_{LVI3}		2.89	2.99	3.09	V
		V_{LVI4}		2.74	2.84	2.94	V
		V_{LVI5}		2.58	2.68	2.78	V
		V_{LVI6}		2.43	2.53	2.63	V
		V_{LVI7}		2.28	2.38	2.48	V
		V_{LVI8}		2.12	2.22	2.32	V
		V_{LVI9}		1.97	2.07	2.17	V
		V_{LVI10}		1.81	1.91	2.01	V
	External input pin[Note 1]	V_{EXLVI}	EXLVI < V_{DD}, 1.8 V ≤ V_{DD} ≤ 3.6 V	1.11	1.21	1.31	V
	Power supply voltage on power application	V_{PUPLVI}	When LVI default start function enabled is set	1.87	2.07	2.27	V
Minimum pulse width		t_{LW}		200			μs
Detection delay time						200	μs
Operation stabilization wait time[Note 2]		t_{LWAIT}				10	μs

Notes 1. The EXLVI/P120/INTP0 pin is used.

2. Time required from setting bit 7 (LVION) of the low-voltage detection register (LVIM) to 1 to operation stabilization

Remark $V_{LVI(n-1)} > V_{LVIn}$: n = 1 to 10

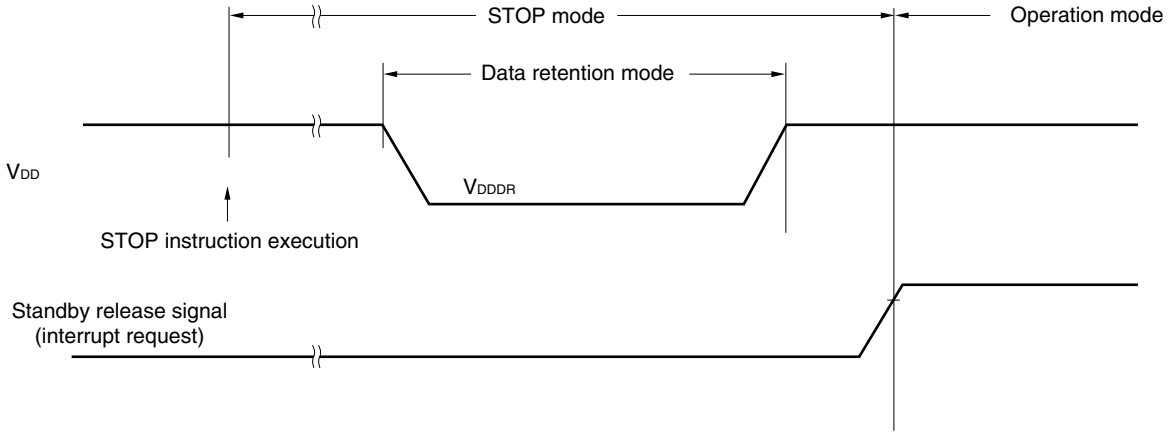
LVI Circuit Timing



Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.5[Note]		3.6	V

Note The value depends on the POC detection voltage. When the voltage drops, the data is retained until a POC reset is effected, but data is not retained when a POC reset is effected.



Flash Memory Programming Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

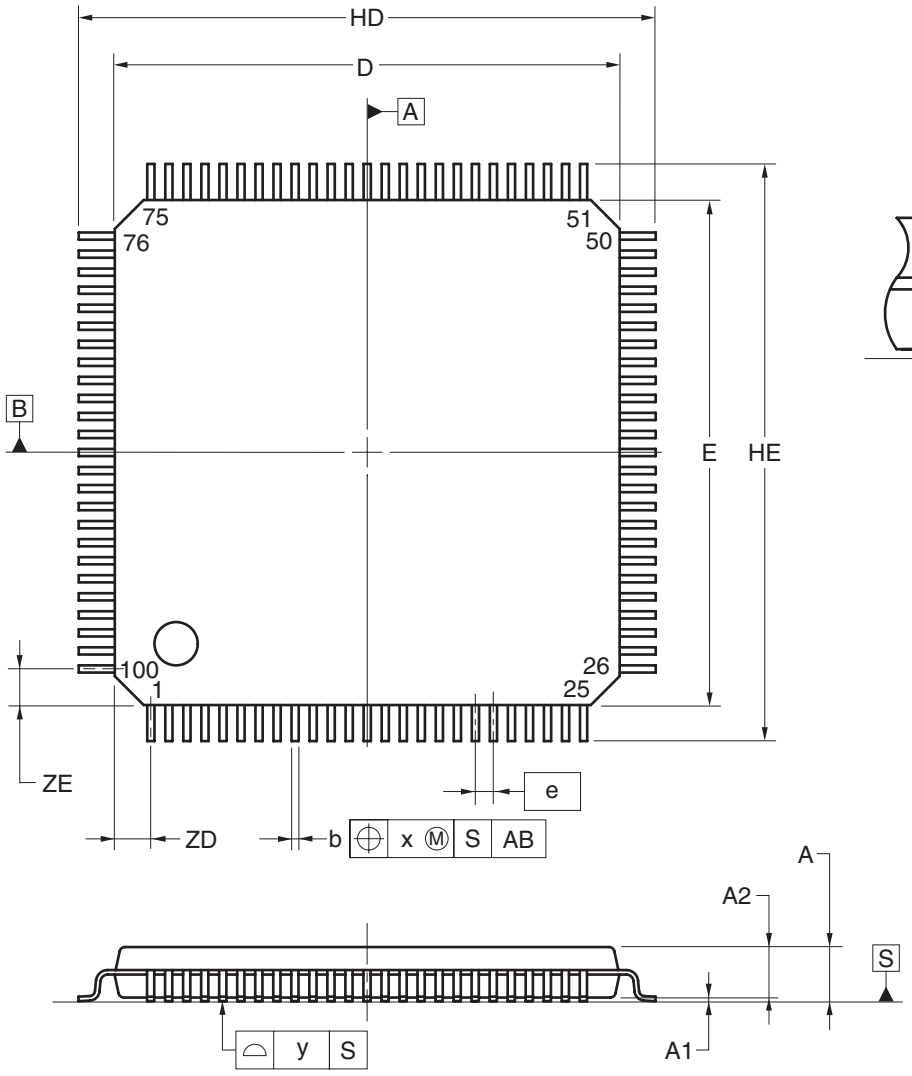
Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
V_{DD} supply current	I_{DD}	Typ. = 10 MHz, Max. = 20 MHz				6	20	mA
Number of rewrites per chip	C_{erwr}	1 erase + 1 write after erase = 1 rewrite[Note]	When a flash memory programmer is used, and the libraries provided by Renesas Electronics are used	Retention: 15 years	1000			Times
			When the EEPROM emulation libraries provided by Renesas Electronics are used	Retention: 5 years	10000			Times

Note When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

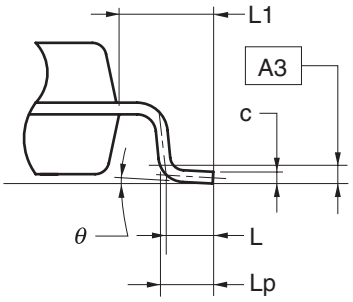
<R>

CHAPTER 33 PACKAGE DRAWING

100-PIN PLASTIC LQFP(FINE PITCH) (14x14)



detail of lead end



(UNIT:mm)

ITEM	DIMENSIONS
D	14.00±0.20
E	14.00±0.20
HD	16.00±0.20
HE	16.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.22±0.05
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.00
ZE	1.00

P100GC-50-GBR

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

(1/2)

Page	Description	Classification
CHAPTER 1 OUTLINE		
p.4	Addition of caution to **1.3 Pin Configuration (Top View)**	(c)
CHAPTER 2 PIN FUNCTIONS		
p.10	Change of **2.1 (1) Port functions**	(c)
p.12 to 14	Change of **2.1 (2) Non-port functions**	(c)
p.28	Change of **2.2.21 $\overline{\text{RESET}}$**	(c)
p.28	Change of **2.2.22 LRESET**	(c)
p.31	Change of **2.2.28 (7) LV_{SS}** and **(10) V_{SS}**	(c)
p.36	Change of TYPE 5-AN in **Figure 2-1. Pin I/O Circuit List (2/4)**	(a)
CHAPTER 3 CPU ARCHITECTURE		
p.59	Change of **Table 3-5. SFR List (5/5)**	(a)
p.70	Change of **Table 3-7. Extended SFR (3rd SFR) List (2/5)**	(a)
CHAPTER 4 PORT FUNCTIONS		
p.88	Change of **Table 4-2. Port functions (1/2)**	(c)
p.123	Addition of **4.2.13 Port L0**	(c)
p.141	Change of **Table 4-5. Settings of Port Mode Register and Output Latch When Using Alternate Function (1/3)**	(a)
CHAPTER 5 CLOCK GENERATOR		
p.145	Addition of caution to **5.1 (1) Main system clock**	(c)
p.198	Addition of caution 5 to **Figure 5-18. Format of Clock Output Select Register 0 (CKS0)**	(c)
CHAPTER 6 TIMER ARRAY UNIT		
p.274	Change of **Figure 6-61. Example of Basic Timing of Operation as One-Shot Pulse Output Function (Start Trigger TImn Input Valid Edge)**	(a)
CHAPTER 7 REAL-TIME COUNTER		
p.282	Change of **Figure 7-2. Format of Real-Time Counter Control Register 0 (RTCC0)**	(a)
p.283	Change of **Figure 7-3. Format of Real-Time Counter Control Register 1 (RTCC1)**	(a)
p.285	Change of **Figure 7-4. Format of Real-Time Counter Control Register 2 (RTCC2)**	(a)
CHAPTER 11 24-BIT ΔΣ-TYPE A/D CONVERTER		
p.372	Change of caution in **11.5 (2)**	(c)
CHAPTER 13 SERIAL INTERFACE IICA		
p.520	Change of **Figure 13-9. Format of IICA Control Register 1 (IICCTL1) (1/2)**	(c)
CHAPTER 14 EXTENDED SFR (3rd SFR) INTERFACE		
p.587	Change of caution in **14.1 Functions of Extended SFR (3rd SFR) Interface**	(c)
p.601	Change of caution in **Figure 14-12. Format of Serial Data Register 02 (SDR02)**	(c)
p.604	Change of caution 1 in **Figure 14-15. Format of Port Mode Registers 1 (PM1)**	(c)
p.608	Change of note in **14.4.2 (1) Overview of functions**	(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

(2/2)

Page	Description	Classification
CHAPTER 15 LCD CONTROLLER/DRIVER		
p.618	Change of note in **Table 15-1. Maximum Number of Pixels**	(c)
p.630, 631	Change of **15.5 (2) Internal voltage boosting method** and **(3) Capacitor split method**	(c)
p.651	Change of **15.8.3 Capacitor split method**	(c)
CHAPTER 18 INTERRUPT FUNCTIONS		
p.699	Change of **Figure 18-6. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (2/2)**	(a)
CHAPTER 19 POWER CALCULATION CIRCUIT		
p.783	Change of **19.4.11 Notes on power calculation circuit (1)**	(c)
CHAPTER 20 POWER QUALITY MEASUREMENT CIRCUIT		
p.794	Change of **Figure 20-6. Format of Peak Current Level Specification Register (IPKLMT)**	(a)
p.794	Change of **Figure 20-7. Format of Peak Voltage Level Specification Register (VPKLMT)**	(a)
CHAPTER 21 DIGITAL FREQUENCY CONVERSION CIRCUIT		
p.813	Change of **Figure 21-7. Pulse Output (Mode 1)**	(c)
CHAPTER 23 RESET FUNCTION		
p.843	Change of **23.2 Cautions after resetting**	(c)
CHAPTER 31 INSTRUCTION SET		
p.914	Change of **Table 31-5. Operation List (10/17)**	(c)
CHAPTER 32 ELECTRICAL SPECIFICATIONS		
p.932-935	Change of **supply current** and **operating current** of **DC Characteristics**	(b), (c)
CHAPTER 33 PACKAGE DRAWING		
p.970	Addition of chapter	(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

78K0R/Lx3-M User's Manual: Hardware

Publication Date: Rev.0.02 Jul 01, 2011
Rev.1.00 Mar 19, 2012

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.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

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: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, 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 Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1

78K0R/Lx3-M

Renesas Electronics Corporation

R01UH0182EJ0100