User's Manual

16

78K0R/Hx3

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

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

Renesas Electronics
www.renesas.com

Rev.3.00 Sep 2013

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

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

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

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

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

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

(2012.4)

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/Hx3 and design and develop application systems and programs for these devices.

- 78K0R/HC3: μPD78F1031, 78F1032, 78F1033, 78F1034, 78F1035
- 78K0R/HE3: μPD78F1036, 78F1037, 78F1038, 78F1039, 78F1040
- 78K0R/HF3: μPD78F1041, 78F1042, 78F1043, 78F1044, 78F1045
- 78K0R/HG3: μPD78F1046, 78F1047, 78F1048, 78F1049, 78F1050

Purpose

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

Organization

The 78K0R/Hx3 manual is separated into two parts: this manual and the instructions edition (common to the 78K0R Microcontroller Series).

78K0R/Hx3 **User's Manual** **(This Manual)**	**78K0R Microcontroller** **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 Microcontroller Instructions User's Manual (U17792E)**.

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/Hx3 User's Manual	This manual
78K0R Microcontroller Instructions User's Manual	U17792E
78K0R Microcontroller Self Programming Library Type02 User's Manual[Note]	U19193E

Note This document is classified under engineering management. Contact an Renesas Electronics sales representative.

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
PM+ Ver. 6.30		U18416E
ID78K0R-QB Ver. 3.20 Integrated Debugger	Operation	U17839E

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

Document Name	Document No.
QB-MINI2 On-Chip Debug Emulator with Programming Function	R20UT0449E
QB-78K0RFX3 In-Circuit Emulator	R20UT0779E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP5 Flash Memory Programmer User's Manual	R02UT0008E

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

Other Documents

Document Name	Document No.
RENESAS MPUs & MCUs RL78 Family	R01CP0003E
Semiconductor Package Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E
Semiconductor Reliability Handbook	R51ZZ0001E

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

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

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

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

○ Minimum instruction execution time can be changed from high speed (42 ns: @ 24 MHz operating frequency) to ultra low-speed (33 μs: @ 30 kHz operation with internal low-speed oscillator)

○ General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)

○ ROM, RAM capacities

ROM	High-Speed RAM	78K0R/HC3	78K0R/HE3	78K0R/HF3	78K0R/HG3
		48 Pins	64 Pins	80 Pins	100 Pins
256 KB	16 KB	μPD78F1035	μPD78F1040	μPD78F1045	μPD78F1050
192 KB	12 KB	μPD78F1034	μPD78F1039	μPD78F1044	μPD78F1049
128 KB	8 KB	μPD78F1033	μPD78F1038	μPD78F1043	μPD78F1048
96 KB	6 KB	μPD78F1032	μPD78F1037	μPD78F1042	μPD78F1047
64 KB	4 KB	μPD78F1031	μPD78F1036	μPD78F1041	μPD78F1046

○ 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)

○ Data protection (with illegal-memory access detection function)

○ Safety (with specific-register protection/prevent unintended occurrences of overflows of the watchdog timer function)

○ On-chip debug function

○ On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)

○ On-chip watchdog timer (operable with the on-chip internal low-speed oscillation clock)

○ On-chip multiplier (16 bits × 16 bits, 32 bits/32 bits), key interrupt function, clock output controller, On-chip BCD adjustment, I/O ports, timer array unit, serial array unit, LIN-UART, CAN controller

○ 10-bit resolution A/D converter (AV_{REF} = 2.7 to 5.5 V)

○ Power supply voltage: V_{DD} = 2.7 to 5.5 V

○ Operating ambient temperature: T_A = –40 to +85°C

Caution The 78K0R/Hx3 has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Remark The functions mounted depend on the product. See **1.6 Block Diagram** and **1.7 Outline of Functions**.

1.2 Applications

- ○ Industrial equipment
- ○ OA equipment
- ○ Home appliances
- ○ Audio visual equipment

1.3 Ordering Information

[List of Part Number]

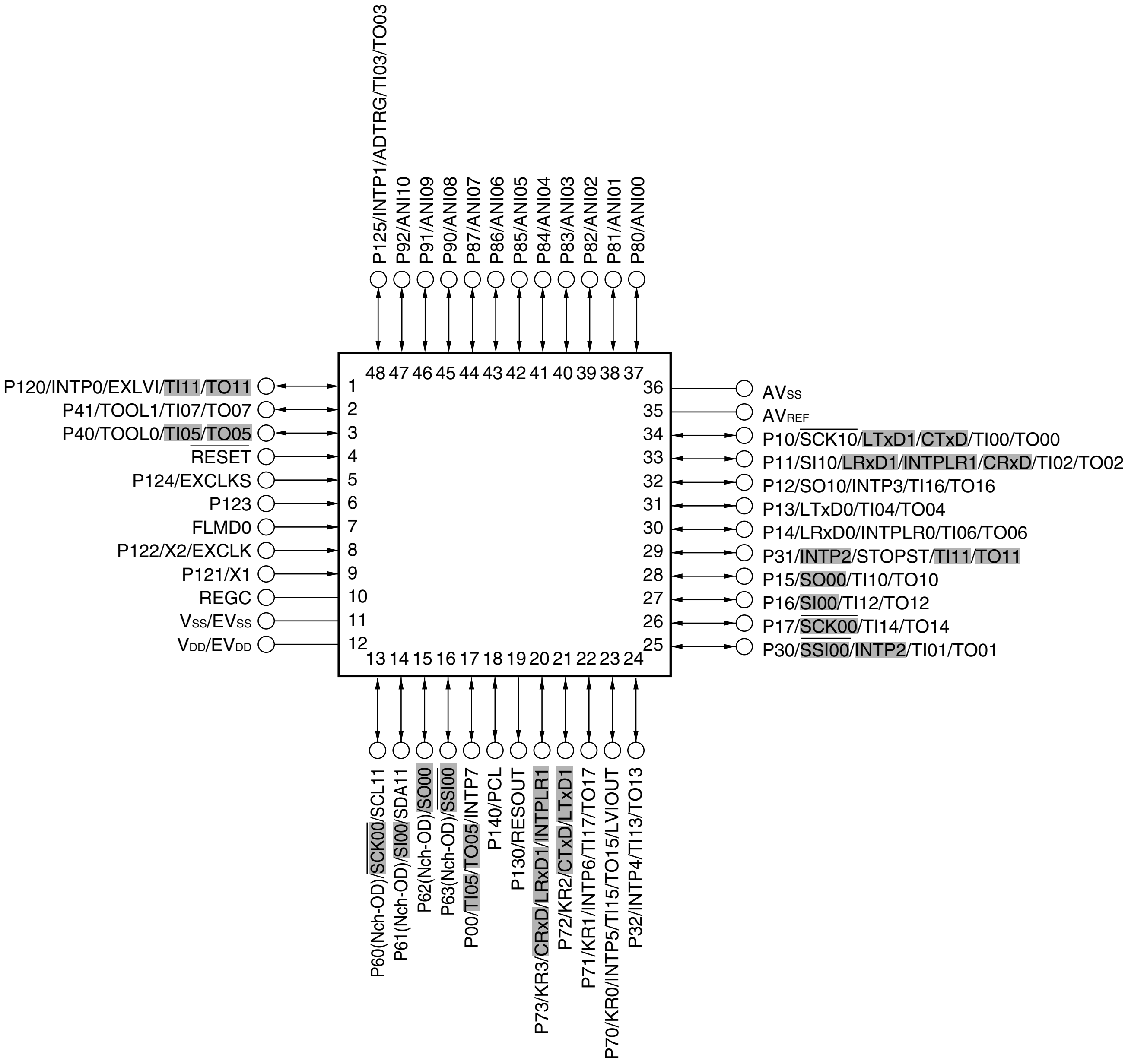
78K0R/Hx3 Microcontrollers	Package	Part Number
78K0R/HC3	48-pin plastic LQFP (fine pitch) (7 × 7)	μPD78F1031GA-GAM-AX, 78F1032GA-GAM-AX, 78F1033GA-GAM-AX, 78F1034GA-GAM-AX, 78F1035GA-GAM-AX
78K0R/HE3	64-pin plastic LQFP (fine pitch) (10 × 10)	μPD78F1036GB-GAH-AX, 78F1037GB-GAH-AX, 78F1038GB-GAH-AX, 78F1039GB-GAH-AX, 78F1040GB-GAH-AX
78K0R/HF3	80-pin plastic LQFP (fine pitch) (12 × 12)	μPD78F1041GK-GAK-AX, 78F1042GK-GAK-AX, 78F1043GK-GAK-AX, 78F1044GK-GAK-AX, 78F1045GK-GAK-AX
78K0R/HG3	100-pin plastic LQFP (fine pitch) (14 × 14)	μPD78F1046GC-UEU-AX, 78F1047GC-UEU-AX, 78F1048GC-UEU-AX, 78F1049GC-UEU-AX, 78F1050GC-UEU-AX

Caution The 78K0R/Hx3 has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

1.4 Pin Configuration (Top View)

1.4.1 78K0R/HC3

- 48-pin plastic LQFP (fine pitch) (7 × 7)



Cautions 1. Make AVSS and EVSS the same potential as VSS.

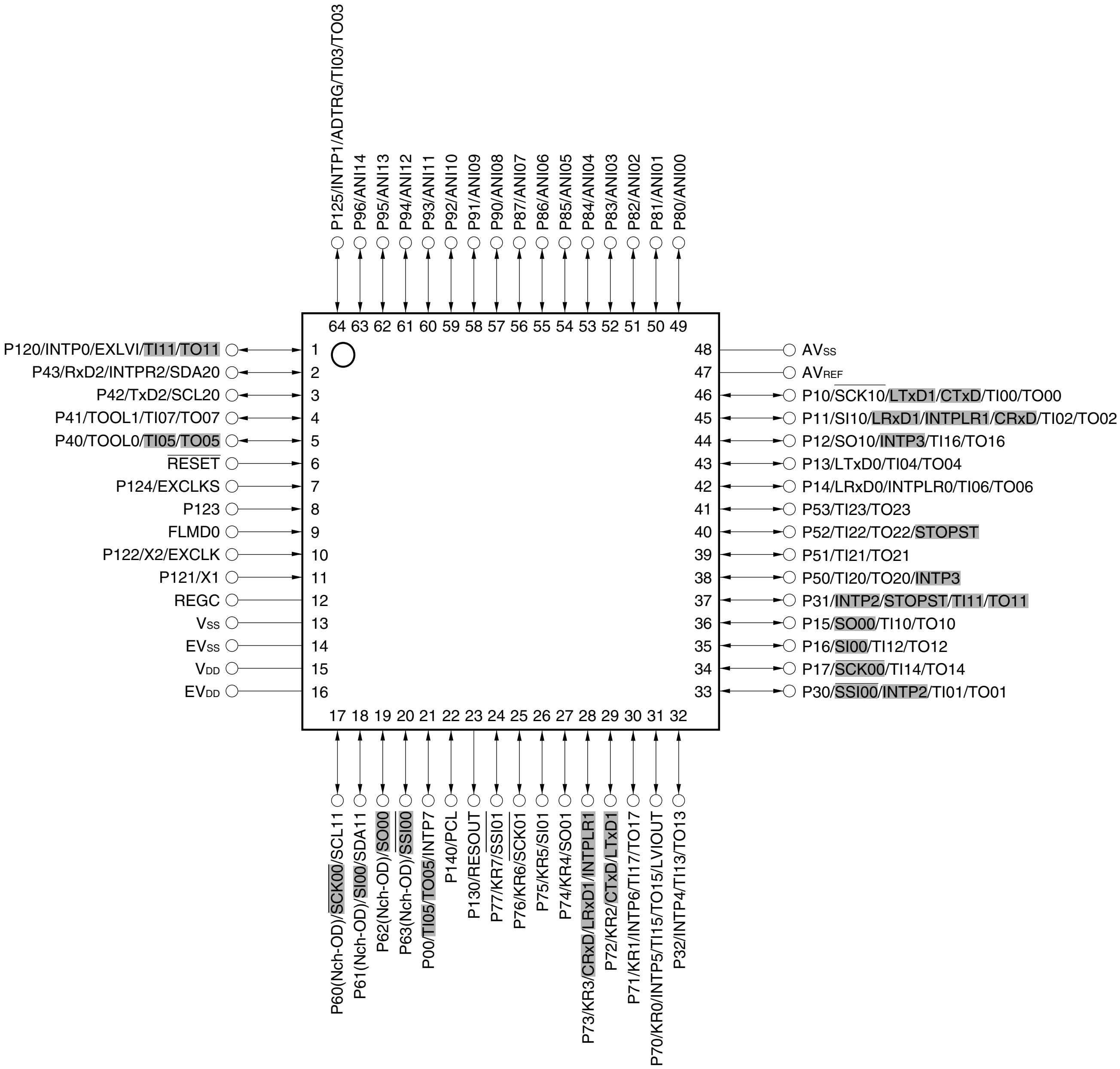
<R> **2. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF).**

3. The shaded pins are provided at two ports. Select either port by using the corresponding register.

4. P80/ANI00 to P87/ANI07 and P90/ANI08 to P92/ANI10 are set as analog inputs in the order of P80/ANI00, …, P87/ANI07, P90/ANI08, …, P92/ANI10 by the A/D port configuration register (ADPC). When using P80/ANI00 to P87/ANI07 and P90/ANI08 to P92/ANI10 as analog inputs, start designing from P80/ANI00 (see 10.3 (8) A/D port configuration register (ADPC) for details).

1.4.2 78K0R/HE3

- 64-pin plastic LQFP (fine pitch) (10 × 10)



Cautions 1. Make AVSS and EVSS the same potential as VSS.

2. Make EVDD the same potential as VDD.

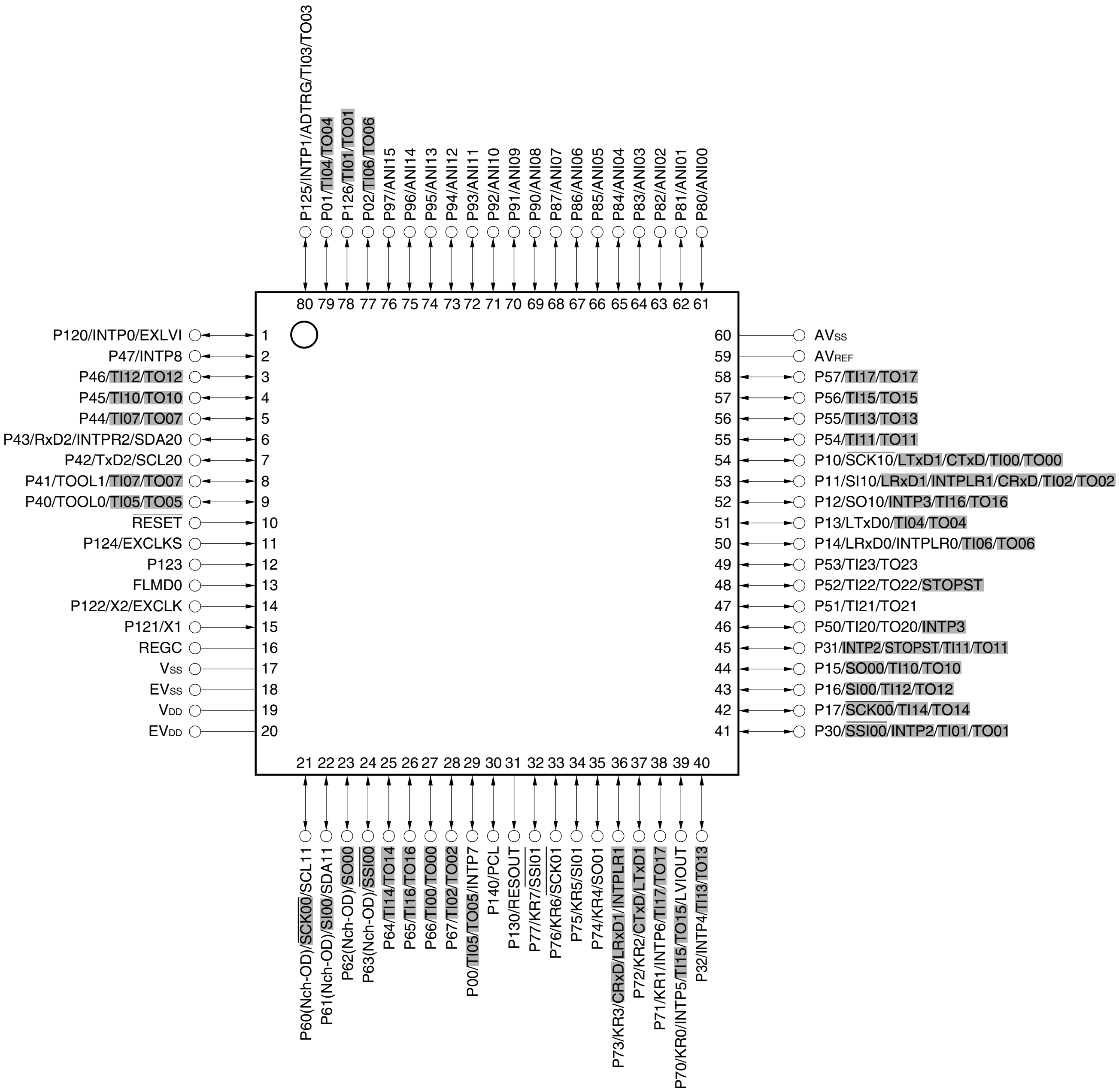
<R> **3. Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).**

4. The shaded pins are provided at two ports. Select either port by using the corresponding register.

5. P80/ANI00 to P87/ANI07 and P90/ANI08 to P96/ANI14 are set as analog inputs in the order of P80/ANI00, ..., P87/ANI07, P90/ANI08, ..., P96/ANI14 by the A/D port configuration register (ADPC). When using P80/ANI00 to P87/ANI07 and P90/ANI08 to P96/ANI14 as analog inputs, start designing from P80/ANI00 (see 10.3 (8) A/D port configuration register (ADPC) for details).

1.4.3 78K0R/HF3

- 80-pin plastic LQFP (fine pitch) (12× 12)



Cautions 1. Make AVSS and EVSS the same potential as VSS.

2. Make EVDD the same potential as VDD.

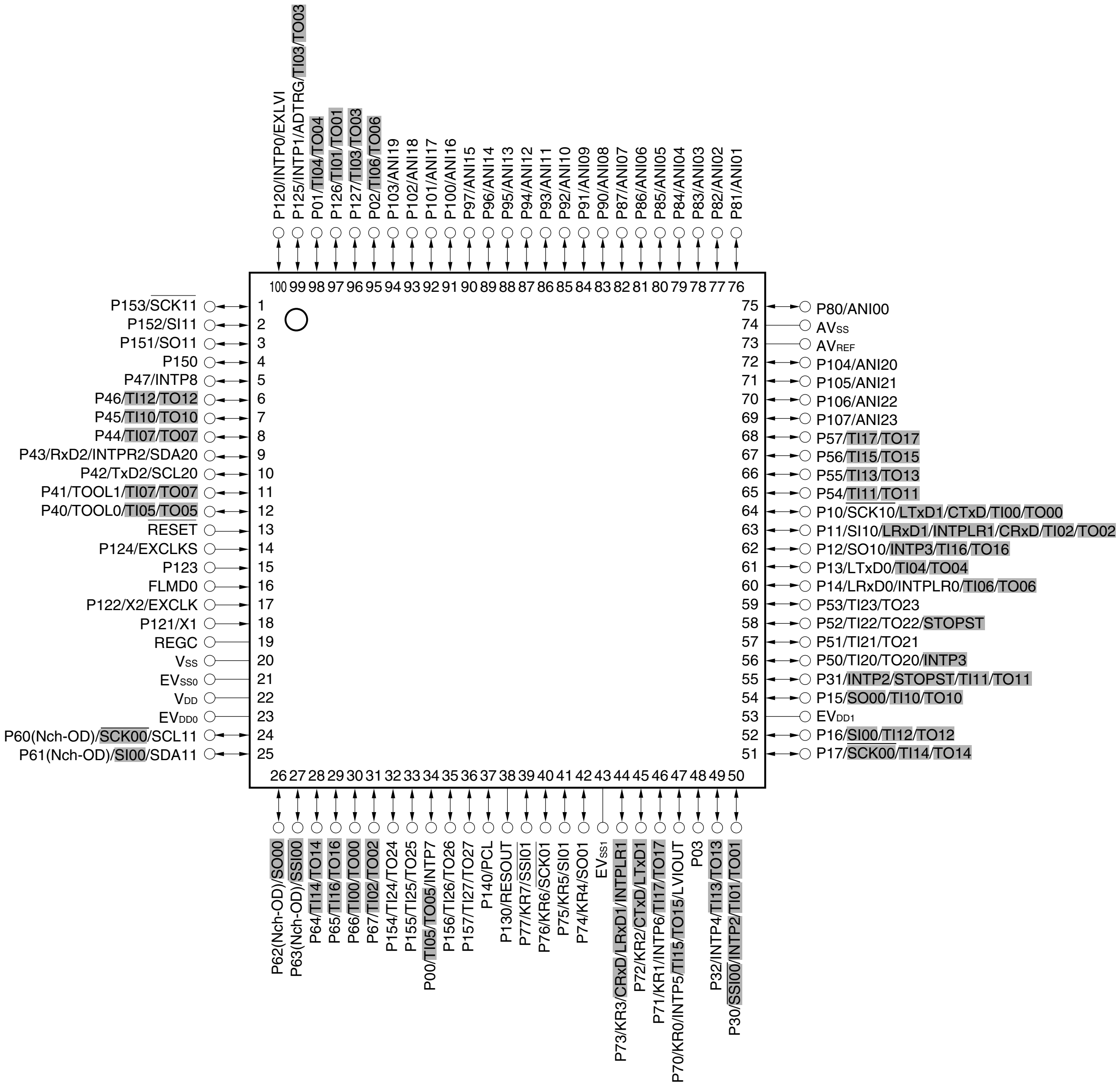
<R> **3. Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).**

4. The shaded pins are provided at two ports. Select either port by using the corresponding register.

5. P80/ANI00 to P87/ANI07 and P90/ANI08 to P97/ANI15 are set as analog inputs in the order of P80/ANI00, ..., P87/ANI07, P90/ANI08, ..., P97/ANI15 by the A/D port configuration register (ADPC). When using P80/ANI00 to P87/ANI07 and P90/ANI08 to P97/ANI15 as analog inputs, start designing from P80/ANI00 (see 10.3 (8) A/D port configuration register (ADPC) for details).

1.4.4 78K0R/HG3

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



Cautions 1. Make AV$_{\mathrm{SS}}$, EV$_{\mathrm{SS0}}$, and EV$_{\mathrm{SS1}}$ the same potential as V$_{\mathrm{SS}}$.

2. Make EV$_{\mathrm{DD0}}$ and EV$_{\mathrm{DD1}}$ the same potential as V$_{\mathrm{DD}}$.

<R> **3. Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).**

4. The shaded pins are provided at two ports. Select either port by using the corresponding register.

5. P80/ANI00 to P87/ANI07, P90/ANI08 to P97/ANI15 and P100/ANI16 to P107/ANI23 set as analog inputs in the order of P80/ANI00, ..., P87/ANI07, P90/ANI08, ..., P97/ANI15, P100/ANI16, ..., P107/ANI23 by the A/D port configuration register (ADPC). When using P80/ANI00 to P87/ANI07, P90/ANI08 to P97/ANI15 and P100/ANI16 to P107/ANI23 as analog inputs, start designing from P80/ANI00 (see 10.3 (8) A/D port configuration register (ADPC) for details).

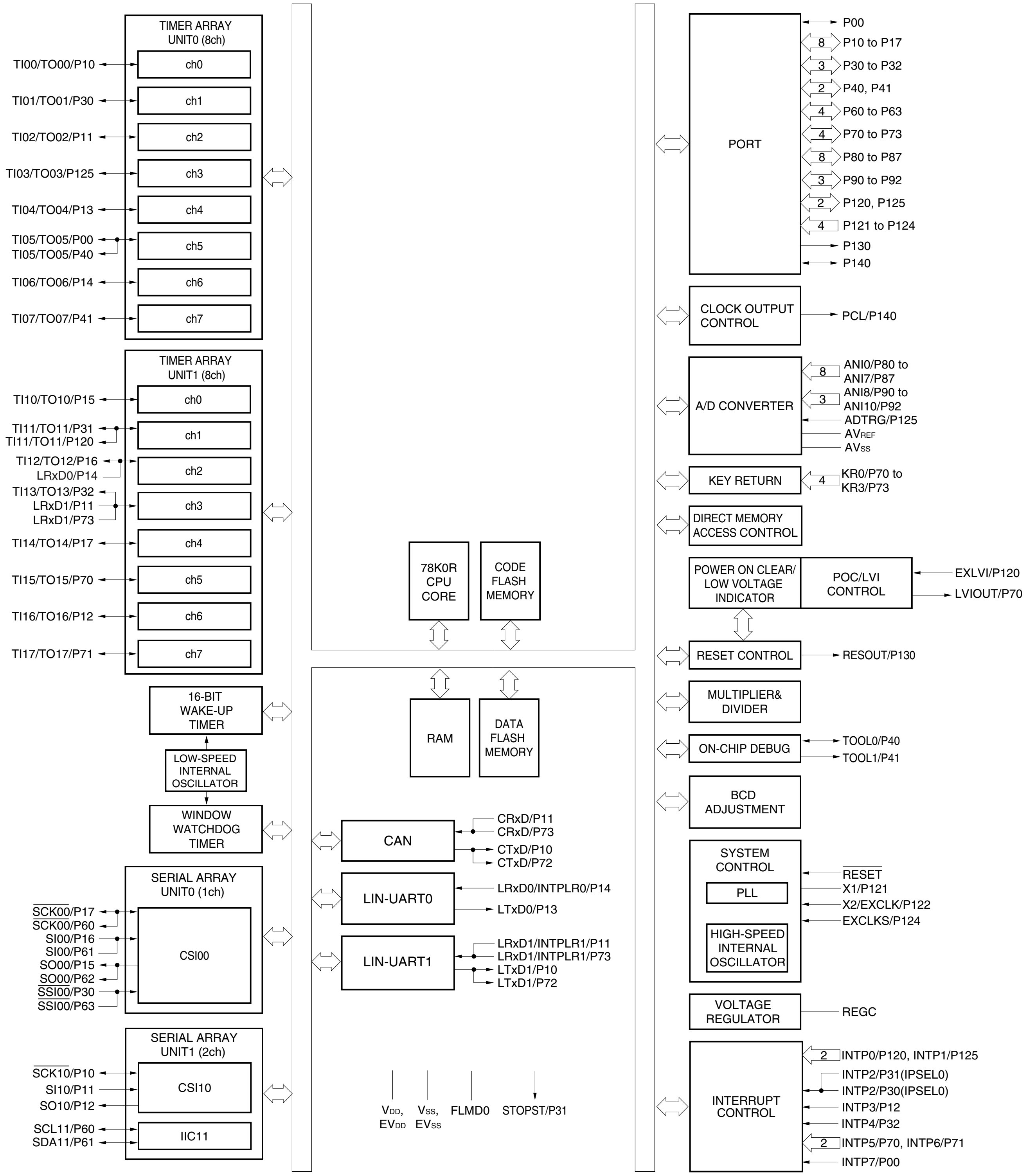
Remark When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to supply separate powers to the V$_{\mathrm{DD}}$ and two EV$_{\mathrm{DD}}$ pins and connect the V$_{\mathrm{SS}}$ and two EV$_{\mathrm{SS}}$ pins to separate ground lines.

1.5 Pin Identification

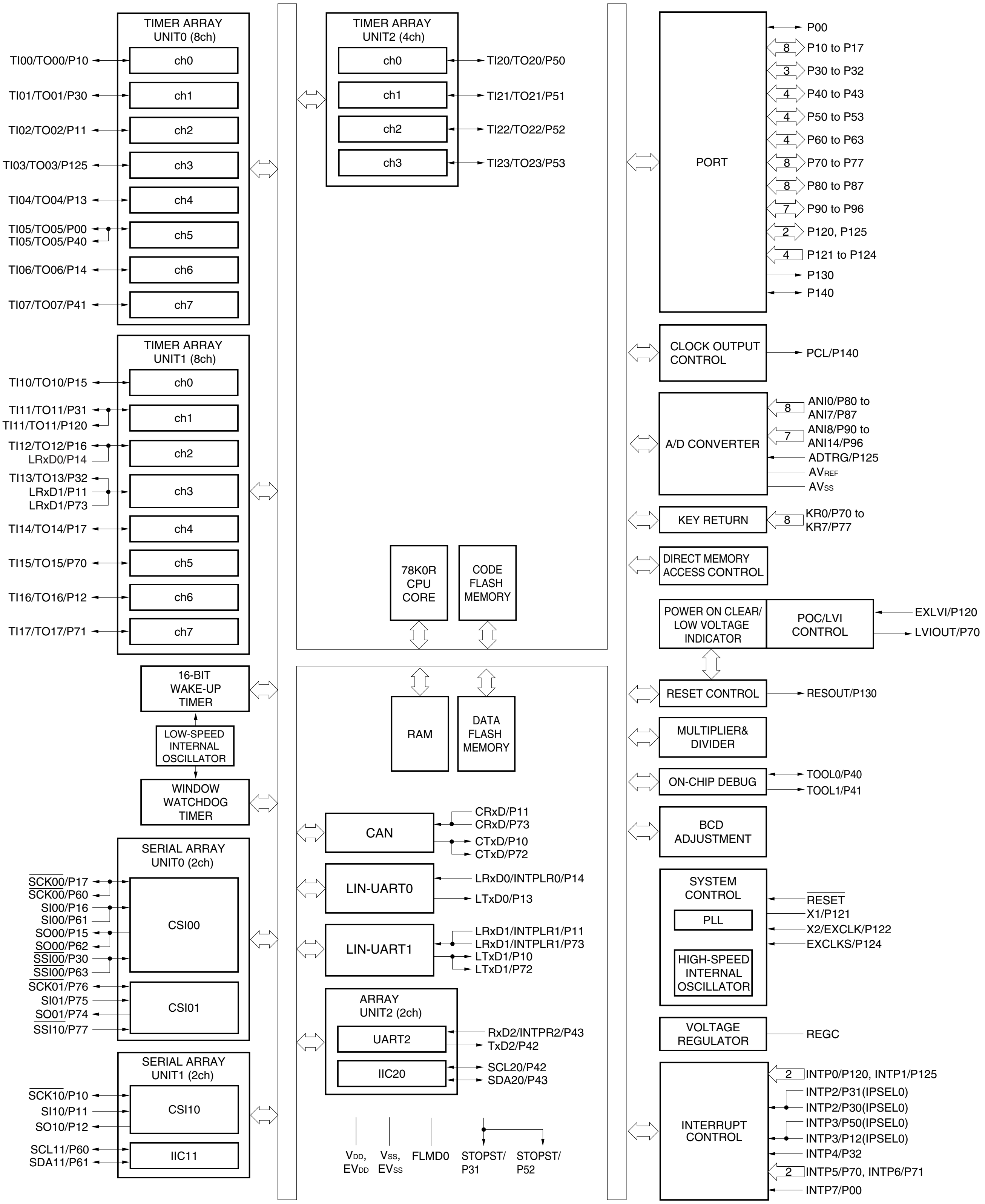
ANI0 to ANI23 :	Analog Input
ADTRG :	AD Trigger
AV_{REF} :	Analog Reference Voltage
AV_{SS} :	Analog Ground
CTxD :	Transmit data for CAN
CRxD :	Receive data for CAN
EV_{DD}, EV_{DD0}, EV_{DD1} :	Power Supply for Port
EV_{SS}, EV_{SS0}, EV_{SS1} :	Ground for Port
EXCLK :	External Clock Input (Main System Clock)
EXCLKS :	External Clock Input (Sub-clock)
EXLVI :	External potential Input for Low-voltage detector
FLMD0 :	Flash Programming Mode
INTP0 to INTP8 :	External Interrupt Input
INTPLR0, INTPLR1 :	External Interrupt Input for LIN-UART
INTPR2 :	External Interrupt Input for UART2
KR0 to KR7 :	Key Return
LRxD :	Receive Data for LIN-UART
LTxD :	Transmit Data for LIN-UART
LVIOUT:	Low-Voltage Detection Flag Output
P00 to P03 :	Port 0
P10 to P17 :	Port 1
P30 to P32 :	Port 3
P40 to P47 :	Port 4
P50 to P57 :	Port 5
P60 to P67 :	Port 6
P70 to P77 :	Port 7
P80 to P87 :	Port 8
P90 to P97 :	Port 9
P100 to P107 :	Port 10
P120 to P127 :	Port 12
P130 :	Port 13
P140 :	Port 14
P150 to P157 :	Port 15
PCL :	Programmable Clock Output
REGC :	Regulator Capacitance
$\overline{\text{RESET}}$:	Reset
RESOUT :	RESET Output
RxD2 :	Receive Data
$\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$, $\overline{\text{SCK11}}$:	Serial Clock Input/Output
SCL11, SCL20 :	Serial Clock Input/Output
SDA11, SDA20 :	Serial Data Input/Output
SI00, SI01, SI10, SI11 :	Serial Data Input
SO00, SO01, SO10, SO11 :	Serial Data Output
$\overline{\text{SSI00}}$, $\overline{\text{SSI01}}$:	Serial Interface Chip Select Input
STOPST :	STOP Status Output
TI00 to TI07, TI10 to TI17, TI20 to TI27 :	Timer Input
TO00 to TO07, TO10 to TO17, TO20 to TO27 :	Timer Output
TOOL0 :	Data Input/Output for Tool
TOOL1 :	Clock Output for Tool
TxD2 :	Transmit Data
V_{DD} :	Power Supply
V_{SS} :	Ground
X1, X2 :	Crystal Oscillator (Main System Clock)

1.6 Block Diagram

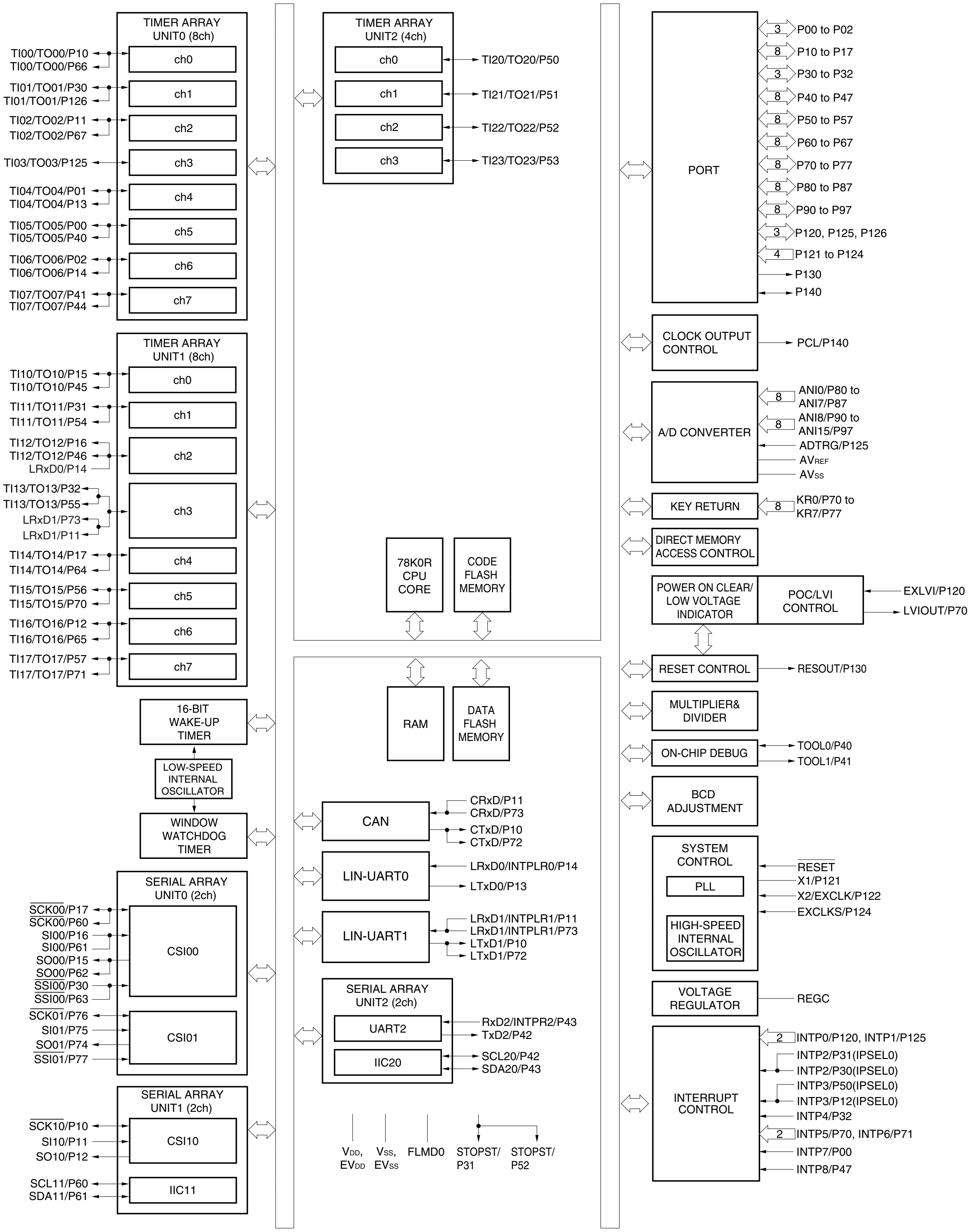
1.6.1 78K0R/HC3



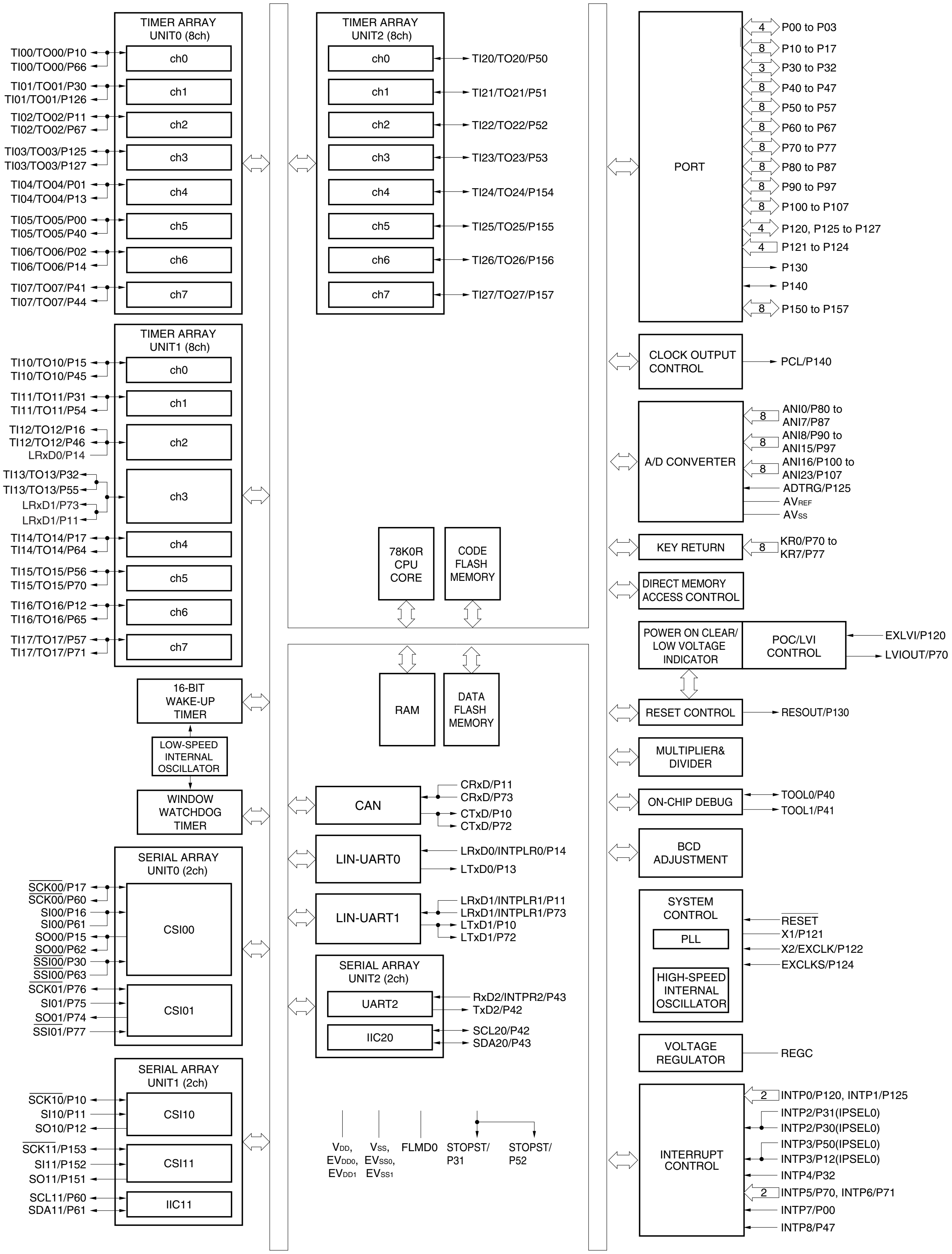
1.6.2 78K0R/HE3



1.6.3 78K0R/HF3



1.6.4 78K0R/HG3



1.7 Outline of Functions

(1/4)

Item		Part Number	78K0R/HC3				
			μPD78F1031	μPD78F1032	μPD78F1033	μPD78F1034	μPD78F1035
Code flash memory			64 KB	96 KB	128 KB	192 KB	256 KB
Data flash memory			16 KB	16 KB	16 KB	16 KB	16 KB
High-Speed RAM			4 KB	6 KB	8 KB	12 KB	16 KB
Regulator			Provided				
Clock	Main	Operating frequency	2 to 24 MHz: V_{DD} = 2.7 to 5.5 V				
		High-speed system	2 to 20 MHz: V_{DD} = 2.7 to 5.5 V				
		Internal high-speed oscillation	4, 8 MHz (TYP.): V_{DD} = 2.7 to 5.5 V				
	PLL oscillation		×1, ×6, ×8				
	Internal low-speed oscillation		30 kHz (TYP.): V_{DD} = 2.7 to 5.5 V				
General-purpose register			8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time			42 ns (24 MHz: V_{DD} = 2.7 to 5.5 V)/50 ns (20 MHz: V_{DD} = 2.7 to 5.5 V)/ 0.1 μs (10 MHz: V_{DD} = 2.7 to 5.5 V)/0.2 μs (5 MHz: V_{DD} = 2.7 to 5.5 V)/ 33 μs (30 kHz : When internal low-speed oscillator is used)				
Instruction set			• 8-bit operation, 16-bit operation • Bit manipulation (Set, reset, test, and Boolean operation), etc.				
I/O port			Total: 41 CMOS I/O: 32 CMOS input: 4 CMOS output: 1 N-ch open-drain I/O (6 V tolerance): 4				
Timer			• 16-bit timer: 16 channels • Watchdog timer: 1 channel • 16-bit wakeup timer: 1 channel				
		Timer outputs	16 (PWM outputs unit 0: 7, unit 1: 7)				
Clock output			Provided				
A/D converter			10-bit resolution × 11 channels				
Serial interface			• CSI: 1 channel (supports full SPI) • CSI: 1 channel • Simplified I^2C: 1 channel • LIN-UART: 2 channels • CAN controller : 1 channel				
Multiplier/divider			• 16 bits × 16 bits = 32 bits (multiplication) • 32 bits ÷ 32 bits = 32 bits, 32-bit remainder (division)				
DMA controller			4 channels				
Interrupt	External		10				
<R>	Internal		40				
Key interrupt			Key interrupt (INTKR) occurs by detecting falling edge of the key input pins (KR0 to KR3).				
Reset			• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by clock monitor • Internal reset by illegal-memory access • Internal reset by power-on-clear • Internal reset by low-voltage detector • Internal reset by illegal instruction execution[Note]				
On-chip debug function			Provided				
Power supply voltage			V_{DD} = 2.7 to 5.5 V				
Operating ambient temperature			T_A = –40 to +85°C				
Package			48-pin plastic LQFP (fine pitch) (7 × 7)				

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.

(2/4)

Part Number Item			78K0R/HE3				
			μPD78F1036	μPD78F1037	μPD78F1038	μPD78F1039	μPD78F1040
Code flash memory			64 KB	96 KB	128 KB	192 KB	256 KB
Data flash memory			16 KB	16 KB	16 KB	16 KB	16 KB
High-Speed RAM			4 KB	6 KB	8 KB	12 KB	16 KB
Regulator			Provided				
Clock	Main	Operating frequency	2 to 24 MHz: V_{DD} = 2.7 to 5.5 V				
		High-speed system	2 to 20 MHz: V_{DD} = 2.7 to 5.5 V				
		Internal high-speed oscillation	4, 8 MHz (TYP.): V_{DD} = 2.7 to 5.5 V				
	PLL oscillation		×1, ×6, ×8				
	Internal low-speed oscillation		30 kHz (TYP.): V_{DD} = 2.7 to 5.5 V				
General-purpose register			8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time			42 ns (24 MHz: V_{DD} = 2.7 to 5.5 V)/50 ns (20 MHz: V_{DD} = 2.7 to 5.5 V)/ 0.1 μs (10 MHz: V_{DD} = 2.7 to 5.5 V)/0.2 μs (5 MHz: V_{DD} = 2.7 to 5.5 V)/ 33 μs (30 kHz : When internal low-speed oscillator is used)				
Instruction set			• 8-bit operation, 16-bit operation • Bit manipulation (Set, reset, test, and Boolean operation), etc.				
I/O port			Total: 55 CMOS I/O: 46 CMOS input: 4 CMOS output: 1 N-ch open-drain I/O (6 V tolerance): 4				
Timer			• 16-bit timer: 20 channels • Watchdog timer: 1 channel • 16-bit wakeup timer: 1 channel				
		Timer outputs	20 (PWM outputs unit 0: 7, unit 1: 7, unit 2: 3)				
Clock output			Provided				
A/D converter			10-bit resolution × 15 channels				
Serial interface			• CSI: 2 channels (supports full SPI) • CSI: 1 channel • Smplified I^2C: 1 channel • UART 1 channel / simplified I^2C: 1 channel • LIN-UART: 2 channels • CAN controller : 1 channel				
Multiplier/divider			• 16 bits × 16 bits = 32 bits (multiplication) • 32 bits ÷ 32 bits = 32 bits, 32-bit remainder (division)				
DMA controller			4 channels				
Interrupt	External		11				
	Internal		47				
Key interrupt			Key interrupt (INTKR) occurs by detecting falling edge of the key input pins (KR0 to KR7).				
Reset			• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by clock monitor • Internal reset by illegal-memory access • Internal reset by power-on-clear • Internal reset by low-voltage detector • Internal reset by illegal instruction execution[Note]				
On-chip debug function			Provided				
Power supply voltage			V_{DD} = 2.7 to 5.5 V				
Operating ambient temperature			T_A = −40 to +85°C				
Package			64-pin plastic LQFP (fine pitch) (10 × 10)				

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.

(3/4)

Part Number			78K0R/HF3				
Item			μPD78F1041	μPD78F1042	μPD78F1043	μPD78F1044	μPD78F1045
Code flash memory			64 KB	96 KB	128 KB	192 KB	256 KB
Data flash memory			16 KB	16 KB	16 KB	16 KB	16 KB
High-Speed RAM			4 KB	6 KB	8 KB	12 KB	16 KB
Regulator			Provided				
Clock	Main	Operating frequency	2 to 24 MHz: V_{DD} = 2.7 to 5.5 V				
		High-speed system	2 to 20 MHz: V_{DD} = 2.7 to 5.5 V				
		Internal high-speed oscillation	4, 8 MHz (TYP.): V_{DD} = 2.7 to 5.5 V				
	PLL oscillation		×1, ×6, ×8				
	Internal low-speed oscillation		30 kHz (TYP.): V_{DD} = 2.7 to 5.5 V				
General-purpose register			8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time			42 ns (24 MHz: V_{DD} = 2.7 to 5.5 V)/50 ns (20 MHz: V_{DD} = 2.7 to 5.5 V)/ 0.1 μs (10 MHz: V_{DD} = 2.7 to 5.5 V)/0.2 μs (5 MHz: V_{DD} = 2.7 to 5.5 V)/ 33 μs (30 kHz : When internal low-speed oscillator is used)				
Instruction set			• 8-bit operation, 16-bit operation • Bit manipulation (Set, reset, test, and Boolean operation), etc.				
I/O port			Total: 71 CMOS I/O: 62 CMOS input: 4 CMOS output: 1 N-ch open-drain I/O (6 V tolerance): 4				
Timer			• 16-bit timer: 20 channels • Watchdog timer: 1 channel • 16-bit wakeup timer: 1 channel				
		Timer outputs	20 (PWM outputs unit 0: 7, unit 1: 7, unit 2: 3)				
Clock output			Provided				
A/D converter			10-bit resolution × 16 channels				
Serial interface			• CSI: 2 channels (supports full SPI) • CSI: 1 channel • Simplified I^2C: 1 channel • UART 1 channel / simplified I^2C: 1 channel • LIN-UART: 2 channels • CAN controller : 1 channel				
Multiplier/divider			• 16 bits × 16 bits = 32 bits (multiplication) • 32 bits ÷ 32 bits = 32 bits, 32-bit remainder (division)				
DMA controller			4 channels				
Interrupt	External		12				
	Internal		47				
Key interrupt			Key interrupt (INTKR) occurs by detecting falling edge of the key input pins (KR0 to KR7).				
Reset			• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by clock monitor • Internal reset by illegal-memory access • Internal reset by power-on-clear • Internal reset by low-voltage detector • Internal reset by illegal instruction execution[Note]				
On-chip debug function			Provided				
Power supply voltage			V_{DD} = 2.7 to 5.5 V				
Operating ambient temperature			T_A = –40 to +85°C				
Package			80-pin plastic LQFP (fine pitch) (12 × 12)				

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.

(4/4)

Item \ Part Number			78K0R/HG3				
			μPD78F1046	μPD78F1047	μPD78F1048	μPD78F1049	μPD78F1050
Code flash memory			64 KB	96 KB	128 KB	192 KB	256 KB
Data flash memory			16 KB	16 KB	16 KB	16 KB	16 KB
High-Speed RAM			4 KB	6 KB	8 KB	12 KB	16 KB
Regulator			Provided				
Clock	Main	Operating frequency	2 to 24 MHz: V_{DD} = 2.7 to 5.5 V				
		High-speed system	2 to 20 MHz: V_{DD} = 2.7 to 5.5 V				
		Internal high-speed oscillation	4, 8 MHz (TYP.): V_{DD} = 2.7 to 5.5 V				
	PLL oscillation		×1, ×6, ×8				
	Internal low-speed oscillation		30 kHz (TYP.): V_{DD} = 2.7 to 5.5 V				
General-purpose register			8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time			42 ns (24 MHz: V_{DD} = 2.7 to 5.5 V)/50 ns (20 MHz: V_{DD} = 2.7 to 5.5 V)/ 0.1 μs (10 MHz: V_{DD} = 2.7 to 5.5 V)/0.2 μs (5 MHz: V_{DD} = 2.7 to 5.5 V)/ 33 μs (30 kHz : When internal low-speed oscillator is used)				
Instruction set			• 8-bit operation, 16-bit operation • Bit manipulation (Set, reset, test, and Boolean operation), etc.				
I/O port			Total: 89 CMOS I/O: 80 CMOS input: 4 CMOS output: 1 N-ch open-drain I/O (6 V tolerance): 4				
Timer			• 16-bit timer: 24 channels • Watchdog timer: 1 channel • 16-bit wakeup timer: 1 channel				
		Timer outputs	24 (PWM outputs unit 0: 7, unit 1: 7, unit 2: 7)				
Clock output			Provided				
A/D converter			10-bit resolution × 24 channels				
Serial interface			• CSI: 2 channels (supports full SPI) • CSI: 1 channel • CSI: 1 channel / Simplified I^{2}C: 1 channel • UART 1 channel / simplified I^{2}C: 1 channel • LIN-UART: 2 channels • CAN controller : 1 channel				
Multiplier/divider			• 16 bits × 16 bits = 32 bits (multiplication) • 32 bits ÷ 32 bits = 32 bits, 32-bit remainder (division)				
DMA controller			4 channels				
Interrupt	External		12				
	Internal		49				
Key interrupt			Key interrupt (INTKR) occurs by detecting falling edge of the key input pins (KR0 to KR7).				
Reset			• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by clock monitor • Internal reset by illegal-memory access • Internal reset by power-on-clear • Internal reset by low-voltage detector • Internal reset by illegal instruction execution[Note]				
On-chip debug function			Provided				
Power supply voltage			V_{DD} = 2.7 to 5.5 V				
Operating ambient temperature			T_A = –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

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

Table 2-1. Pin I/O Buffer Power Supplies (AV_{REF}, EV_{DD}, V_{DD})

- 78K0R/HC3: 48-pin plastic LQFP (fine pitch) (7 × 7)
- 78K0R/HE3: 64-pin plastic LQFP (fine pitch) (10 × 10)
- 78K0R/HF3: 80-pin plastic LQFP (fine pitch) (12 × 12)

Power Supply	Corresponding Pins
AV_{REF}	P80 to P87, P90 to P97
EV_{DD}	Port pins other than P80 to P87, P90 to P97, and P121 to P124
V_{DD}	• P121 to P124 • Pins other than port pins

Table 2-2. Pin I/O Buffer Power Supplies (AV_{REF}, EV_{DD0}, EV_{DD1}, V_{DD})

- 78K0R/HG3: 100-pin plastic LQFP (fine pitch) (14 × 14)

Power Supply	Corresponding Pins
AV_{REF}	P80 to P87, P90 to P97, P100 to P107
EV_{DD0}, EV_{DD1}	Port pins other than P80 to P87, P90 to P97, P100 to P107, and P121 to P124
V_{DD}	• P121 to P124 • Pins other than port pins

2.1.1 78K0R/HC3

(1) Port functions (1/2): 78K0R/HC3

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 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	INTP7/TI05/TO05
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI00/SCK10/TO00/ LTxD1/CTxD
P11				TI02/SI10/LRxD1/ INTPLR1/TO02/ CRxD
P12				INTP3/TI16/SO10/ TO16
P13				TI04/LTxD0/TO04
P14				TI06/LRxD0/INTPLR0/ TO06
P15				TI10/SO00/TO10
P16				TI12/SI00/TO12
P17				TI14/SCK00/TO14
P30	I/O	Port 3. 3-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	INTP2/SSI00/TI01/ TO01
P31				INTP2/TI11/STOPST/ TO11
P32				INTP4/TI13/TO13
P40	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/TI05/TO05
P41				TOOL1/TI07/TO07
P60	I/O	Port 6. 4-bit I/O port. Output of P60 to P63 is N-ch open-drain output (6 V tolerance). Input of P60, P61 and P63 can be set to TTL input buffer[Note]. 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	SCK00/SCL11
P61				SI00/SDA11
P62				SO00
P63				SSI00
P70	I/O	Port 7. 4-bit I/O port. Input of P73 can be set to TTL input buffer[Note]. Output of P72 can be set to N-ch open-drain output (V_{DD} tolerance)[Note]. 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	INTP5/KR0/TI15/TO15/ LVIOUT
P71				INTP6/KR1/TI17/TO17
P72				KR2/CTxD/LTxD1
P73				KR3/CRxD/LRxD1/ INTPLR1

Note For details, see **4.4.4 Connecting to external device with different power potential (3 V)**.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(1) Port functions (2/2): 78K0R/HC3

Function Name	I/O	Function	After Reset	Alternate Function
P80 to P87	I/O	Port 8. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI00 to ANI07
P90 to P92	I/O	Port 9. 3-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI08 to ANI10
P120	I/O	Port 12. 2-bit I/O port and 4-bit input port. Input/output can be specified in 1-bit units. For only P120 and P125, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI/TI11/TO11
P121	Input			X1
P122				X2/EXCLK
P123				−
P124				EXCLKS
P125	I/O			INTP1/ADTRG/TI03/TO03
P130	Output	Port 13. 1-bit output port.	Output port	RESOUT
P140	I/O	Port 14. 1-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting.	Output port	PCL

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (1/4): 78K0R/HC3

Function Name	I/O	Function	After Reset	Alternate Function
ANI00 to ANI10	Input	A/D converter analog input	Digital input port	P80 to P87, P90 to P92
ADTRG	Input	A/D converter external trigger input	Input port	P125/INTP1/TI03/TO03
CRxD	Input	CAN receive data input	Input port	P11/SI10/LRxD1/INTPLR1/TI02/TO02, P73/KR3/LRxD1/INTPLR1
CTxD	Output	CAN transmit data output	Input port	P10/SCK10/LTxD1/TI00/TO00, P72/KR2/LTxD1
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0/TI11/TO11
FLMD0	–	Flash memory programming mode setting	–	–
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/TI11/TO11
INTP1				P125/ADTRG/TI03/TO03
INTP2				P30/SSI00/TI01/TO01, P31/STOPST/TI11/TO11
INTP3				P12/SO10/TI16/TO16
INTP4				P32/TI13/TO13
INTP5				P70/KR0/TI15/TO15/LVIOUT
INTP6				P71/KR1/TI17/TO17
INTP7				P00/TI05/TO05
INTPLR0	Input	External interrupt request input for which the valid edge for LIN-UART0 (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P14/LRxD0/TI06/TO06
INTPLR1	Input	External interrupt request input for which the valid edge for LIN-UART1 (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P11/TI02/SI10/LRxD1/CRxD/TO02, P73/KR3/CRxD/LRxD1
KR0	Input	Key interrupt input	Input port	P70/INTP5/TI15/TO15/LVIOUT
KR1				P71/INTP6/TI17/TO17
KR2				P72/CTxD/LTxD1
KR3				P73/CRxD/LRxD1/INTPLR1
LRxD0	Input	Serial data input to LIN-UART0	Input port	P14/INTPLR0/TI06/TO06
LRxD1	Input	Serial data input to LIN-UART1	Input port	P11/TI02/SI10/INTPLR1/CRxD/TO02, P73/CRxD/KR3/INTPLR1

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (2/4): 78K0R/HC3

Function Name	I/O	Function	After Reset	Alternate Function
LTxD0	Output	Serial data output from LIN-UART0	Input port	P13/TI04/TO04
LTxD1	Output	Serial data output from LIN-UART1	Input port	P10/$\overline{\text{SCK10}}$/CTxD/ TI00/TO00, P72/CTxD/KR2
LVIOUT	Output	Low-voltage detection flag output	Input port	P70/INTP5/KR0/TI15/ TO15
PCL	Output	Clock output	Output port	P140
REGC	–	Connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF). <R>	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
RESOUT	Output	RESET output	Output port	P130
$\overline{\text{SCK00}}$	I/O	Clock input/output for CSI00 and CSI10	Input port	P17/TI14/TO14, P60/SCL11
$\overline{\text{SCK10}}$				P10/LTxD1/CTxD/ TI00/TO00
SCL11	I/O	Clock input/output for simplified I^2C	Input port	P60/$\overline{\text{SCK00}}$
SDA11	I/O	Serial data I/O for simplified I^2C	Input port	P61/SI00
SI00	Input	Serial data input to CSI00 and CSI10	Input port	P16/TI12/TO12, P61/SDA11
SI10				P11/LRxD1/CRxD/ INTPLR1/TI02/TO02
SO00	Output	Serial data output from CSI00 and CSI10	Input port	P15/TI10/TO10, P62
SO10				P12/INTP3/TI16/TO16
$\overline{\text{SSI00}}$	Input	Chip select input to CSI00	Input port	P30/INTP2/TI01/TO01, P63
STOPST	Output	STOP status output	Input port	P31/INTP2/TI11/TO11
TI00	Input	External count clock input to 16-bit timer 00	Input port	P10/$\overline{\text{SCK10}}$/LTxD1/ CTxD/TO00
TI01		External count clock input to 16-bit timer 01		P30/$\overline{\text{SSI00}}$/INTP2/ TO01
TI02		External count clock input to 16-bit timer 02		P11/SI10/LRxD1/ CRxD/INTPLR1/TO02
TI03		External count clock input to 16-bit timer 03		P125/INTP1/ADTRG/ TO03
TI04		External count clock input to 16-bit timer 04		P13/LTxD0/TO04
TI05		External count clock input to 16-bit timer 05		P40/TOOL0/TO05, P00/INTP7/TO05
TI06		External count clock input to 16-bit timer 06		P14/LRxD0/INTPLR0/ TO06
TI07		External count clock input to 16-bit timer 07		P41/TOOL1/TO07
TI10		External count clock input to 16-bit timer 10		P15/SO00/TO10

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (3/4): 78K0R/HC3

Function Name	I/O	Function	After Reset	Alternate Function
TI11	Input	External count clock input to 16-bit timer 11	Input port	P31/INTP2/STOPST/ TO11, P120/INTP0/EXLVI/ TO11
TI12		External count clock input to 16-bit timer 12		P16/SI00/TO12
TI13		External count clock input to 16-bit timer 13		P32/INTP4/TO13
TI14		External count clock input to 16-bit timer 14		P17/SCK00/TO14
TI15		External count clock input to 16-bit timer 15		P70/KR0/INTP5/TO15/ LVIOUT
TI16		External count clock input to 16-bit timer 16		P12/SO10/INTP3/TO16
TI17		External count clock input to 16-bit timer 17		P71/KR1/INTP6/TO17
TO00	Output	16-bit timer 00 output	Input port	P10/SCK10/LTxD1/ CTxD/TI00
TO01		16-bit timer 01 output		P30/SSI00/INTP2/TI01
TO02		16-bit timer 02 output		P11/SI10/LRxD1/ CRxD/INTPLR1/TI02
TO03		16-bit timer 03 output		P125/INTP1/ADTRG/ TI03
TO04		16-bit timer 04 output		P13/LTxD0/TI04
TO05		16-bit timer 05 output		P40/TOOL0/TI05, P00/INTP7/TI05
TO06		16-bit timer 06 output		P14/LRxD0/INTPLR0/ TI06
TO07		16-bit timer 07 output		P41/TOOL1/TI07
TO10		16-bit timer 10 output		P15/SO00/TI10
TO11		16-bit timer 11 output		P31/INTP2/STOPST/ TI11, P120/INTP0/EXLVI/ TI11
TO12		16-bit timer 12 output		P16/SI00/TI12
TO13		16-bit timer 13 output		P32/INTP4/TI13
TO14		16-bit timer 14 output		P17/SCK00/TI14
TO15		16-bit timer 15 output		P70/KR0/INTP5/TI15/ LVIOUT
TO16		16-bit timer 16 output		P12/SO10/INTP3/TI16
TO17		16-bit timer 17 output		P71/KR1/INTP6/TI17
TOOL0	I/O	Data I/O for flash memory programmer/debugger	Input port	P40/TI05/TO05
TOOL1	Output	Clock output for debugger	Input port	P41/TI07/TO07
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
EXCLKS	Input	External clock input for subclock	Input port	P124

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (4/4): 78K0R/HC3

Function Name	I/O	Function	After Reset	Alternate Function
V_{DD}	–	Positive power supply (P121 to P124 and other than ports (other than $\overline{\text{RESET}}$ pin and FLMD0 pin))	–	–
EV_{DD}	–	Positive power supply for ports (other than P80 to P87, P90 to P92, and P121 to P124), and $\overline{\text{RESET}}$ and FLMD0 pin. Make EV_{DD} the same potential as V_{DD}.	–	–
AV_{REF}	–	• A/D converter and comparator reference voltage input • Positive power supply for P80 to P87, P90 to P92, and A/D converter	–	–
V_{SS}	–	Ground potential (P121 to P124 and other than ports (other than $\overline{\text{RESET}}$ pin and FLMD0 pin))	–	–
EV_{SS}	–	Ground potential for ports (other than P80 to P87, P90 to P92, and P121 to P124), and $\overline{\text{RESET}}$ and FLMD0 pin. Make EV_{SS} the same potential as V_{SS}.	–	–
AV_{SS}	–	Ground potential for A/D converter, P80 to P87, and P90 to P92. Make AV_{SS} the same potential as V_{SS} and EV_{SS}.	–	–

2.1.2 78K0R/HE3

(1) Port functions (1/2): 78K0R/HE3

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 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	INTP7/TI05/TO05
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI00/$\overline{\text{SCK10}}$/TO00/ LTxD1/CTxD
P11				TI02/SI10/LRxD1/ INTPLR1/TO02/CRxD
P12				INTP3/TI16/SO10/ TO16
P13				TI04/LTxD0/TO04
P14				TI06/LRxD0/INTPLR0/ TO06
P15				TI10/SO00/TO10
P16				TI12/SI00/TO12
P17				TI14/$\overline{\text{SCK00}}$/TO14
P30	I/O	Port 3. 3-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	INTP2/$\overline{\text{SSI00}}$/TI01/ TO01
P31				INTP2/TI11/STOPST/ TO11
P32				INTP4/TI13/TO13
P40	I/O	Port 4. 4-bit I/O port. Output of P42 and P43 can be set to N-ch open-drain output (V_{DD} tolerance)[Note]. 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/TI05/TO05
P41				TOOL1/TI07/TO07
P42				TxD2/SCL20
P43				RxD2/SDA20/INTPR2
P50	I/O	Port 5. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP3/TI20/TO20
P51				TI21/TO21
P52				TI22/STOPST/TO22
P53				TI23/TO23
P60	I/O	Port 6. 4-bit I/O port. Output of P60 to P63 is N-ch open-drain output (6 V tolerance). Input of P60, P61 and P63 can be set to TTL input buffer[Note]. 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	$\overline{\text{SCK00}}$/SCL11
P61				SI00/SDA11
P62				SO00
P63				$\overline{\text{SSI00}}$

Note For details, see **4.4.4 Connecting to external device with different power potential (3 V)**.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(1) Port functions (2/2): 78K0R/HE3

Function Name	I/O	Function	After Reset	Alternate Function
P70	I/O	Port 7. 8-bit I/O port. Input of P73, P75 to P77 can be set to TTL input buffer[Note]. Output of P72, P74, and P76 can be set to N-ch open-drain output (V_{DD} tolerance)[Note]. 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	INTP5/KR0/TI15/TO15/LVIOUT
P71				INTP6/KR1/TI17/TO17
P72				KR2/CTxD/LTxD1
P73				KR3/CRxD/LRxD1/INTPLR1
P74				KR4/SO01
P75				KR5/SI01
P76				KR6/$\overline{\text{SCK01}}$
P77				KR7/$\overline{\text{SSI01}}$
P80 to P87	I/O	Port 8. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI00 to ANI07
P90 to P96	I/O	Port 9. 7-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI08 to ANI14
P120	I/O	Port 12. 2-bit I/O port and 4-bit input port. Input/output can be specified in 1-bit units. For only P120 and P125, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI/TI11/TO11
P121	Input			X1
P122				X2/EXCLK
P123				–
P124				EXCLKS
P125	I/O			INTP1/ADTRG/TI03/TO03
P130	Output	Port 13. 1-bit output port.	Output port	RESOUT
P140	I/O	Port 14. 1-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting.	Output port	PCL

Note For details, see **4.4.4 Connecting to external device with different power potential (3 V)**.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (1/4): 78K0R/HE3

Function Name	I/O	Function	After Reset	Alternate Function
ANI00 to ANI14	Input	A/D converter analog input	Digital input port	P80 to P87, P90 to P96
ADTRG	Input	A/D converter external trigger input	Input port	P125/INTP1/TI03/TO03
CRxD	Input	CAN receive data input	Input port	P11/SI10/LRxD1/INTPLR1/TI02/TO02, P73/KR3/LRxD1/INTPLR1
CTxD	Output	CAN transmit data output	Input port	P10/SCK10/LTxD1/TI00/TO00, P72/KR2/LTxD1
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0/TI11/TO11
FLMD0	–	Flash memory programming mode setting	–	–
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/TI11/TO11
INTP1				P125/ADTRG/TI03/TO03
INTP2				P30/SSI00/TI01/TO01, P31/STOPST/TI11/TO11
INTP3				P12/SO10/TI16/TO16, P50/TI20/TO20
INTP4				P32/TI13/TO13
INTP5				P70/KR0/TI15/TO15/LVIOUT
INTP6				P71/KR1/TI17/TO17
INTP7				P00/TI05/TO05
INTPLR0	Input	External interrupt request input for which the valid edge for LIN-UART0 (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P14/LRxD0/TI06/TO06
INTPLR1	Input	External interrupt request input for which the valid edge for LIN-UART1 (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P11/TI02/SI10/CRxD/LRxD1/TO02, P73/KR3/CRxD/LRxD1
INTPR2	Input	External interrupt request input for UART2	Input port	P43/RxD2/SDA20
KR0	Input	Key interrupt input	Input port	P70/INTP5/TI15/TO15/LVIOUT
KR1				P71/INTP6/TI17/TO17
KR2				P72/CTxD/LTxD1
KR3				P73/CRxD/LRxD1/INTPLR1
KR4				P74/SO01
KR5				P75/SI01
KR6				P76/SCK01
KR7				P77/SSI01

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (2/4): 78K0R/HE3

Function Name	I/O	Function	After Reset	Alternate Function
LRxD0	Input	Serial data input to LIN-UART0	Input port	P14/INTPLR0/TI06/TO06
LRxD1	Input	Serial data input to LIN-UART1	Input port	P11/TI02/SI10/CRxD/INTPLR1/TO02, P73/CRxD/KR3/INTPLR1
LTxD0	Output	Serial data output from LIN-UART0	Input port	P13/TI04/TO04
LTxD1	Output	Serial data output from LIN-UART1	Input port	P10/TI00/$\overline{\text{SCK10}}$/CTxD/TO00, P72/CTxD/KR2
LVIOUT	Output	Low-voltage detection flag output	Input port	P70/INTP5/KR0/TI15/TO15
PCL	Output	Clock output	Output port	P140
<R> REGC	–	Connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF).	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
RESOUT	Output	RESET output	Output port	P130
RxD2	Input	Serial data input to UART2	Input port	P43/INTPR2/SDA20
SCK00	I/O	Clock input/output for CSI00, CSI01, and CSI10	Input port	P17/TI14/TO14, P60/SCL11
$\overline{\text{SCK01}}$				P76/KR6
$\overline{\text{SCK10}}$				P10/LTxD1/CTxD/TI00/TO00
SCL11	I/O	Clock input/output for simplified I^2C	Input port	P60/SCK00
SCL20				P42/TxD2
SDA11	I/O	Serial data I/O for simplified I^2C	Input port	P61/SI00
SDA20				P43/RxD2/INTPR2
SI00	Input	Serial data input to CSI00, CSI01, and CSI10	Input port	P16/TI12/TO12, P61/SDA11
SI01				P75/KR5
SI10				P11/LRxD1/CRxD/INTPLR1/TI02/TO02
SO00	Output	Serial data output from CSI00, CSI01, and CSI10	Input port	P15/TI10/TO10, P62
SO01				P74/KR4
SO10				P12/INTP3/TI16/TO16
SSI00	Input	Chip select input to CSI00, CSI01	Input port	P30/INTP2/TI01/TO01, P63
$\overline{\text{SSI01}}$				P77/KR7

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (3/4): 78K0R/HE3

Function Name	I/O	Function	After Reset	Alternate Function
STOPST	Output	STOP status output	Input port	P31/INTP2/TI11/TO11, P52/TI22/TO22
TI00	Input	External count clock input to 16-bit timer 00	Input port	P10/SCK10/LTxD1/ CTxD/TO00
TI01		External count clock input to 16-bit timer 01		P30/SSI00/INTP2/TO01
TI02		External count clock input to 16-bit timer 02		P11/SI10/LRxD1/ CRxD/INTPLR1/TO02
TI03		External count clock input to 16-bit timer 03		P125/INTP1/ADTRG/ TO03
TI04		External count clock input to 16-bit timer 04		P13/LTxD0/TO04
TI05		External count clock input to 16-bit timer 05		P40/TOOL0/TO05, P00/INTP7/TO05
TI06		External count clock input to 16-bit timer 06		P14/LRxD0/INTPLR0/ TO06
TI07		External count clock input to 16-bit timer 07		P41/TOOL1/TO07
TI10		External count clock input to 16-bit timer 10		P15/SO00/TO10
TI11		External count clock input to 16-bit timer 11		P31/INTP2/STOPST/ TO11, P120/INTP0/EXLVI/ TO11
TI12		External count clock input to 16-bit timer 12		P16/SI00/TO12
TI13		External count clock input to 16-bit timer 13		P32/INTP4/TO13
TI14		External count clock input to 16-bit timer 14		P17/SCK00/TO14
TI15		External count clock input to 16-bit timer 15		P70/KR0/INTP5/TO15/ LVIOUT
TI16		External count clock input to 16-bit timer 16		P12/SO10/INTP3/TO16
TI17		External count clock input to 16-bit timer 17		P71/KR1/INTP6/TO17
TI20		External count clock input to 16-bit timer 20		P50/TO20/INTP3
TI21		External count clock input to 16-bit timer 21		P51/TO21
TI22		External count clock input to 16-bit timer 22		P52/TO22/STOPST
TI23		External count clock input to 16-bit timer 23		P53/TO23
TO00	Output	16-bit timer 00 output	Input port	P10/SCK10/LTxD1/ CTxD/TI00
TO01		16-bit timer 01 output		P30/SSI00/INTP2/TI01
TO02		16-bit timer 02 output		P11/SI10/LRxD1/ CRxD/INTPLR1/TI02
TO03		16-bit timer 03 output		P125/INTP1/ADTRG/ TI03
TO04		16-bit timer 04 output		P13/LTxD0/TI04

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (4/4): 78K0R/HE3

Function Name	I/O	Function	After Reset	Alternate Function
TO05	Output	16-bit timer 05 output	Input port	P40/TOOL0/TI05, P00/INTP7/TI05
TO06		16-bit timer 06 output		P14/LRxD0/INTPLR0/TI06
TO07		16-bit timer 07 output		P41/TOOL1/TI07
TO10		16-bit timer 10 output		P15/SO00/TI10
TO11		16-bit timer 11 output		P31/INTP2/STOPST/TI11, P120/INTP0/EXLVI/TI11
TO12		16-bit timer 12 output		P16/SI00/TI12
TO13		16-bit timer 13 output		P32/INTP4/TI13
TO14		16-bit timer 14 output		P17/SCK00/TI14
TO15		16-bit timer 15 output		P70/KR0/INTP5/TI15/LVIOUT
TO16		16-bit timer 16 output		P12/SO10/INTP3/TI16
TO17		16-bit timer 17 output		P71/KR1/INTP6/TI17
TO20		16-bit timer 20 output		P50/TI20/INTP3
TO21		16-bit timer 21 output		P51/TI21
TO22		16-bit timer 22 output		P52/TI22/STOPST
TO23		16-bit timer 23 output		P53/TI23
TOOL0	I/O	Data I/O for flash memory programmer/debugger	Input port	P40/TI05/TO05
TOOL1	Output	Clock output for debugger	Input port	P41/TI07/TO07
TxD2	Output	Serial data output from UART2	Input port	P42/SCL20
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
EXCLKS	Input	External clock input for subclock	Input port	P124
V_{DD}	–	Positive power supply (P121 to P124 and other than ports (other than $\overline{\text{RESET}}$ pin and FLMD0 pin))	–	–
EV_{DD}	–	Positive power supply for ports (other than P80 to P87, P90 to P96, and P121 to P124), and $\overline{\text{RESET}}$ and FLMD0 pin. Make EV_{DD} the same potential as V_{DD}.	–	–
AV_{REF}	–	• A/D converter and comparator reference voltage input • Positive power supply for P80 to P87, P90 to P96, and A/D converter	–	–
V_{SS}	–	Ground potential (P121 to P124 and other than ports (other than $\overline{\text{RESET}}$ pin and FLMD0 pin))	–	–
EV_{SS}	–	Ground potential for ports (other than P80 to P87, P90 to P96, and P121 to P124), and $\overline{\text{RESET}}$ and FLMD0 pin. Make EV_{SS} the same potential as V_{SS}.	–	–
AV_{SS}	–	Ground potential for A/D converter, P80 to P87, and P90 to P96. Make AV_{SS} the same potential as V_{SS} and EV_{SS}.	–	–

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

2.1.3 78K0R/HF3

(1) Port functions (1/2): 78K0R/HF3

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 3-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	INTP7/TI05/TO05
P01				TI04/TO04
P02				TI06/TO06
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI00/$\overline{\text{SCK10}}$/TO00/ LTxD1/CTxD
P11				TI02/SI10/LRxD1/ CRxD/INTPLR1/ TO02
P12				INTP3/TI16/SO10/ TO16
P13				TI04/LTxD0/TO04
P14				TI06/LRxD0/INTPLR0/ TO06
P15				TI10/SO00/TO10
P16				TI12/SI00/TO12
P17				TI14/$\overline{\text{SCK00}}$/TO14
P30	I/O	Port 3. 3-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	INTP2/$\overline{\text{SSI00}}$/TI01/ TO01
P31				INTP2/TI11/STOPST/ TO11
P32				INTP4/TI13/TO13
P40	I/O	Port 4. 8-bit I/O port. Output of P42 and P43 can be set to N-ch open-drain output (V_{DD} tolerance)[Note]. 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/TI05/TO05
P41				TOOL1/TI07/TO07
P42				TxD2/SCL20
P43				RxD2/SDA20/INTPR2
P44				TI07/TO07
P45				TI10/TO10
P46				TI12/TO12
P47				INTP8
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	INTP3/TI20/TO20
P51				TI21/TO21
P52				TI22/STOPST/TO22
P53				TI23/TO23
P54				TI11/TO11
P55				TI13/TO13
P56				TI15/TO15
P57				TI17/TO17

Note For details, see **4.4.4 Connecting to external device with different power potential (3 V)**.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(1) Port functions (2/2): 78K0R/HF3

Function Name	I/O	Function	After Reset	Alternate Function
P60	I/O	Port 6. 8-bit I/O port. Output of P60 to P63 is N-ch open-drain output (6 V tolerance). Input of P60, P61 and P63 can be set to TTL input buffer[Note]. 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	SCK00/SCL11
P61				SI00/SDA11
P62				SO00
P63				SSI00
P64				TI14/TO14
P65				TI16/TO16
P66				TI00/TO00
P67				TI02/TO02
P70	I/O	Port 7. 8-bit I/O port. Input of P73, P75 to P77 can be set to TTL input buffer[Note]. Output of P72, P74, and P76 can be set to N-ch open-drain output (V_{DD} tolerance)[Note]. 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	INTP5/KR0/TI15/TO15/ LVIOUT
P71				INTP6/KR1/TI17/TO17
P72				KR2/CTxD/LTxD1
P73				KR3/CRxD/LRxD1/ INTPLR1
P74				KR4/SO01
P75				KR5/SI01
P76				KR6/SCK01
P77				KR7/SSI01
P80 to P87	I/O	Port 8. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI00 to ANI07
P90 to P97	I/O	Port 9. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI08 to ANI15
P120	I/O	Port 12. 3-bit I/O port and 4-bit input port. Input/output can be specified in 1-bit units. For only P120 and P125, P126, 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
P123				–
P124				EXCLKS
P125	I/O			INTP1/ADTRG/TI03/ TO03
P126				TI01/TO01
P130	Output	Port 13. 1-bit output port.	Output port	RESOUT
P140	I/O	Port 14. 1-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting.	Output port	PCL

Note For details, see **4.4.4 Connecting to external device with different power potential (3 V)**.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (1/5): 78K0R/HF3

Function Name	I/O	Function	After Reset	Alternate Function
ANI00 to ANI15	Input	A/D converter analog input	Digital input port	P80 to P87, P90 to P97
ADTRG	Input	A/D converter external trigger input	Input port	P125/INTP1/TI03/TO03
CRxD	Input	CAN receive data input	Input port	P11/TI02/SI10/LRxD1/ INTPLR1/TO02, P73/KR3/LRxD1/ INTPLR1
CTxD	Output	CAN transmit data output	Input port	P10/TI00/SCK10/ LTxD1/TO00, P72/KR2/LTxD1
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0
FLMD0	–	Flash memory programming mode setting	–	–
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
INTP1				P125/ADTRG/TI03/ TO03
INTP2				P30/SSI00/TI01/TO01, P31/STOPST/TI11/ TO11
INTP3				P12/SO10/TI16/TO16, P50/TI20/TO20
INTP4				P32/TI13/TO13
INTP5				P70/KR0/TI15/TO15/ LVIOUT
INTP6				P71/KR1/TI17/TO17
INTP7				P00/TI05/TO05
INTP8				P47
INTPLR0	Input	External interrupt request input for which the valid edge for LIN-UART0 (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P14/LRxD0/TI06/TO06
INTPLR1	Input	External interrupt request input for which the valid edge for LIN-UART1 (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P11/TI02/SI10/LRxD1/ CRxD/TO02, P73/KR3/CRxD/ LRxD1
INTPR2	Input	External interrupt request input for UART2	Input port	P43/RxD2/SDA20
KR0	Input	Key interrupt input	Input port	P70/INTP5/TI15/TO15/ LVIOUT
KR1				P71/INTP6/TI17/TO17
KR2				P72/CTxD/LTxD1
KR3				P73/CRxD/LRxD1/ INTPLR1
KR4				P74/SO01
KR5				P75/SI01
KR6				P76/SCK01
KR7				P77/SSI01
LRxD0	Input	Serial data input to LIN-UART0	Input port	P14/INTPLR0/TI06/ TO06

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (2/5): 78K0R/HF3

Function Name	I/O	Function	After Reset	Alternate Function
LRxD1	Input	Serial data input to LIN-UART1	Input port	P11/TI02/SI10/ INTPLR1/CRxD/TO02, P73/CRxD/KR3/ INTPLR1
LTxD0	Output	Serial data output from LIN-UART0	Input port	P13/TI04/TO04
LTxD1	Output	Serial data output from LIN-UART1	Input port	P10/TI00/$\overline{\text{SCK10}}$/ CTxD/TO00, P72/CTxD/KR2
LVIOUT	Output	Low-voltage detection flag output	Input port	P70/INTP5/KR0/TI15/ TO15
PCL	Output	Clock output	Output port	P140
REGC	–	Connecting regulator output (2.5 V) stabilization capacitance for internal operation. <R> Connect to V_{SS} via a capacitor (0.47 to 1 μF).	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
RESOUT	Output	RESET output	Output port	P130
RxD2	Input	Serial data input to UART2	Input port	P43/INTPR2/SDA20
SCK00	I/O	Clock input/output for CSI00, CSI01, and CSI10	Input port	P17/TI14/TO14, P60/SCL11
$\overline{\text{SCK01}}$				P76/KR6
$\overline{\text{SCK10}}$				P10/LTxD1/CTxD/ TI00/TO00
SCL11	I/O	Clock input/output for simplified I^2C	Input port	P60/SCK00
SCL20				P42/TxD2
SDA11	I/O	Serial data I/O for simplified I^2C	Input port	P61/SI00
SDA20				P43/RxD2/INTPR2
SI00	Input	Serial data input to CSI00, CSI01, and CSI10	Input port	P16/TI12/TO12, P61/SDA11
SI01				P75/KR5
SI10				P11/LRxD1/INTPLR1/ CRxD/TI02/TO02
SO00	Output	Serial data output from CSI00, CSI01, and CSI10	Input port	P15/TI10/TO10, P62
SO01				P74/KR4
SO10				P12/INTP3/TI16/TO16
SSI00	Input	Chip select input to CSI00, CSI01	Input port	P30/INTP2/TI01/TO01, P63
$\overline{\text{SSI01}}$				P77/KR7
STOPST	Output	STOP status output	Input port	P31/INTP2/TI11/TO11, P52/TI22/TO22

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (3/5): 78K0R/HF3

Function Name	I/O	Function	After Reset	Alternate Function
TI00	Input	External count clock input to 16-bit timer 00	Input port	P10/SCK10/LTxD1/ CTxD/TO00, P66/TO00
TI01		External count clock input to 16-bit timer 01		P30/SSI00/INTP0/ TO01, P126/TO01
TI02		External count clock input to 16-bit timer 02		P11/SI10/LRxD1/ INTPLR1/CRxD/ TO02, P67/TO02
TI03		External count clock input to 16-bit timer 03		P125/INTP1/ADTRG/ TO03
TI04		External count clock input to 16-bit timer 04		P13/LTxD0/TO04, P01/TO04
TI05		External count clock input to 16-bit timer 05		P40/TOOL0/TO05, P00/INTP7/TO05
TI06		External count clock input to 16-bit timer 06		P14/LRxD0/INTPLR0/ TO06, P02/TO06
TI07		External count clock input to 16-bit timer 07		P41/TOOL1/TO07, P44/TO07
TI10		External count clock input to 16-bit timer 10		P15/SO00/TO10, P45/TO10
TI11		External count clock input to 16-bit timer 11		P31/INTP2/STOPST/ TO11, P54/TO11
TI12		External count clock input to 16-bit timer 12		P16/SI00/TO12, P46/TO12
TI13		External count clock input to 16-bit timer 13		P32/INTP4/TO13, P55/TO13
TI14		External count clock input to 16-bit timer 14		P17/SCK00/TO14, P64/TO14
TI15		External count clock input to 16-bit timer 15		P70/KR0/INTP5/TO15/ LVIOUT, P56/TO15
TI16		External count clock input to 16-bit timer 16		P12/SO10/INTP3/TO16, P65/TO16
TI17		External count clock input to 16-bit timer 17		P71/KR1/INTP6/TO17, P57/TO17
TI20		External count clock input to 16-bit timer 20		P50/TO20/INTP3
TI21		External count clock input to 16-bit timer 21		P51/TO21
TI22		External count clock input to 16-bit timer 22		P52/TO22/STOPST
TI23		External count clock input to 16-bit timer 23		P53/TO23

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (4/5): 78K0R/HF3

Function Name	I/O	Function	After Reset	Alternate Function
TO00	Output	16-bit timer 00 output	Input port	P10/SCK10/LTxD1/CTxD/TI00, P66/TI00
TO01		16-bit timer 01 output		P30/SSI00/INTP2/TI01, P126/TI01
TO02		16-bit timer 02 output		P11/SI10/LRxD1/CRxD/INTPLR1/TI02, P67/TI02
TO03		16-bit timer 03 output		P125/INTP1/ADTRG/TI03
TO04		16-bit timer 04 output		P13/LTxD0/TI04, P01/TI04
TO05		16-bit timer 05 output		P40/TOOL0/TI05, P00/INTP7/TI05
TO06		16-bit timer 06 output		P14/LRxD0/INTPLR0/TI06, P02/TI06
TO07		16-bit timer 07 output		P41/TOOL1/TI07, P44/TI07
TO10		16-bit timer 10 output		P15/SO00/TI10, P45/TI10
TO11		16-bit timer 11 output		P31/INTP2/STOPST/TI11, P54/TI11
TO12		16-bit timer 12 output		P16/SI00/TI12, P46/TI12
TO13		16-bit timer 13 output		P32/INTP4/TI13, P55/TI13
TO14		16-bit timer 14 output		P17/SCK00/TI14, P64/TI14
TO15		16-bit timer 15 output		P70/KR0/INTP5/TI15/LVIOUT, P56/TI15
TO16		16-bit timer 16 output		P12/SO10/INTP3/TI16, P65/TI16
TO17		16-bit timer 17 output		P71/KR1/INTP6/TI17, P57/TI17
TO20		16-bit timer 20 output		P50/TI20/INTP3
TO21		16-bit timer 21 output		P51/TI21
TO22		16-bit timer 22 output		P52/TI22/STOPST
TO23		16-bit timer 23 output		P53/TI23
TOOL0	I/O	Data I/O for flash memory programmer/debugger	Input port	P40/TI05/TO05
TOOL1	Output	Clock output for debugger	Input port	P41/TI07/TO07
TxD2	Output	Serial data output from UART2	Input port	P42/SCL20

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (5/5): 78K0R/HF3

Function Name	I/O	Function	After Reset	Alternate Function
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
EXCLKS	Input	External clock input for subclock	Input port	P124
V_{DD}	–	Positive power supply (P121 to P124 and other than ports (other than $\overline{\text{RESET}}$ pin and FLMD0 pin))	–	–
EV_{DD}	–	Positive power supply for ports (other than P80 to P87, P90 to P97, and P121 to P124), and $\overline{\text{RESET}}$ and FLMD0 pin. Make EV_{DD} the same potential as V_{DD}.	–	–
AV_{REF}	–	• A/D converter and comparator reference voltage input • Positive power supply for P80 to P87, P90 to P97, and A/D converter	–	–
V_{SS}	–	Ground potential (P121 to P124 and other than ports (other than $\overline{\text{RESET}}$ pin and FLMD0 pin))	–	–
EV_{SS}	–	Ground potential for ports (other than P80 to P87, P90 to P97, and P121 to P124), and $\overline{\text{RESET}}$ and FLMD0 pin. Make EV_{SS} the same potential as V_{SS}.	–	–
AV_{SS}	–	Ground potential for A/D converter, P80 to P87, and P90 to P97. Make AV_{SS} the same potential as V_{SS} and EV_{SS}.	–	–

2.1.4 78K0R/HG3

(1) Port functions (1/3): 78K0R/HG3

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP7/TI05/TO05
P01				TI04/TO04
P02				TI06/TO06
P03				–
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI00/$\overline{\text{SCK10}}$/TO00/ CTxD/LTxD1
P11				TI02/SI10/LRxD1/ INTPLR1/CRxD/TO02
P12				INTP3/TI16/SO10/ TO16
P13				TI04/LTxD0/TO04
P14				TI06/LRxD0/INTPLR0/ TO06
P15				TI10/SO00/TO10
P16				TI12/SI00/TO12
P17				TI14/$\overline{\text{SCK00}}$/TO14
P30	I/O	Port 3. 3-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	INTP2/$\overline{\text{SSI00}}$/TI01/ TO01
P31				INTP2/TI11/STOPST/ TO11
P32				INTP4/TI13/TO13
P40	I/O	Port 4. 8-bit I/O port. Output of P42 and P43 can be set to N-ch open-drain output (V_{DD} tolerance)[Note]. 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/TI05/TO05
P41				TOOL1/TI07/TO07
P42				TxD2/SCL20
P43				RxD2/SDA20/INTPR2
P44				TI07/TO07
P45				TI10/TO10
P46				TI12/TO12
P47				INTP8
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	INTP3/TI20/TO20
P51				TI21/TO21
P52				TI22/STOPST/TO22
P53				TI23/TO23
P54				TI11/TO11
P55				TI13/TO13
P56				TI15/TO15
P57				TI17/TO17

Note For details, see **4.4.4 Connecting to external device with different power potential (3 V)**.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(1) Port functions (2/3): 78K0R/HG3

Function Name	I/O	Function	After Reset	Alternate Function
P60	I/O	Port 6. 8-bit I/O port. Output of P60 to P63 is N-ch open-drain output (6 V tolerance). Input of P60, P61 and P63 can be set to TTL input buffer[Note]. 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	SCK00/SCL11
P61				SI00/SDA11
P62				SO00
P63				SSI00
P64				TI14/TO14
P65				TI16/TO16
P66				TI00/TO00
P67				TI02/TO02
P70	I/O	Port 7. 8-bit I/O port. Input of P73, P75 to P77 can be set to TTL input buffer[Note]. Output of P72, P74, and P76 can be set to N-ch open-drain output (V_{DD} tolerance) [Note]. 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	INTP5/KR0/TI15/TO15/ LVIOUT
P71				INTP6/KR1/TI17/TO17
P72				KR2/CTxD/LTxD1
P73				KR3/CRxD/LRxD1/ INTPLR1
P74				KR4/SO01
P75				KR5/SI01
P76				KR6/$\overline{\text{SCK01}}$
P77				KR7/$\overline{\text{SSI01}}$
P80 to P87	I/O	Port 8. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI00 to ANI07
P90 to P97	I/O	Port 9. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI08 to ANI15
P100 to P107	I/O	Port 10. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI16 to ANI23
P120	I/O	Port 12. 4-bit I/O port and 4-bit input port. Input/output can be specified in 1-bit units. For only P120 and P125 to P127, 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
P123				–
P124				EXCLKS
P125	I/O			INTP1/ADTRG/TI03/ TO03
P126				TI01/TO01
P127				TI03/TO03
P130	Output	Port 13. 1-bit output port.	Output port	RESOUT
P140	I/O	Port 14. 1-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting.	Output port	PCL

Note For details, see **4.4.4 Connecting to external device with different power potential (3 V)**.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(1) Port functions (3/3): 78K0R/HG3

Function Name	I/O	Function	After Reset	Alternate Function
P150	I/O	Port 15. 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	–
P151				SO11
P152				SI11
P153				$\overline{\text{SCK11}}$
P154				TI24/TO24
P155				TI25/TO25
P156				TI26/TO26
P157				TI27/TO27

(2) Non-port functions (1/5): 78K0R/HG3

Function Name	I/O	Function	After Reset	Alternate Function
ANI00 to ANI23	Input	A/D converter analog input	Digital input port	P80 to P87, P90 to P97, P100 to P107
ADTRG	Input	A/D converter external trigger input	Input port	P125/INTP1/TI03/TO03
CRxD	Input	CAN receive data input	Input port	P11/TI02/SI10/ LRxD1/INTPLR1/ TO02, P73/KR3/LRxD1/ INTPLR1
CTxD	Output	CAN transmit data output	Input port	P10/$\overline{\text{SCK10}}$/LTxD1/ TI00/TO00, P72/KR2/LTxD1
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0
FLMD0	–	Flash memory programming mode setting	–	–
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
INTP1				P125/ADTRG/TI03/ TO03
INTP2				P30/$\overline{\text{SSI00}}$/TI01/TO01, P31/STOPST/TI11/ TO11
INTP3				P12/SO10/TI16/TO16, P50/TI20/TO20
INTP4				P32/TI13/TO13
INTP5				P70/KR0/TI15/TO15/ LVIOUT
INTP6				P71/KR1/TI17/TO17
INTP7				P00/TI05/TO05
INTP8				P47
INTPLR0	Input	External interrupt request input for which the valid edge for LIN-UART0 (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P14/LRxD0/TI06/TO06
INTPLR1				P11/TI02/SI10/LRxD1/ CRxD/TO02, P73/KR3/CRxD/LRxD1
INTPR2	Input	External interrupt request input for UART2	Input port	P43/RxD2/SDA20

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (2/5): 78K0R/HG3

Function Name	I/O	Function	After Reset	Alternate Function
KR0	Input	Key interrupt input	Input port	P70/INTP5/TI15/TO15/ LVIOUT
KR1				P71/INTP6/TI17/TO17
KR2				P72/CTxD/LTxD1
KR3				P73/CRxD/LRxD1/ INTPLR1
KR4				P74/SO01
KR5				P75/SI01
KR6				P76/$\overline{\text{SCK01}}$
KR7				P77/$\overline{\text{SSI01}}$
LRxD0	Input	Serial data input to LIN-UART0	Input port	P14/INTPLR0/TI06/ TO06
LRxD1	Input	Serial data input to LIN-UART1	Input port	P11/TI02/SI10/ INTPLR1/CRxD/TO02, P73/CRxD/KR3/ INTPLR1
LTxD0	Output	Serial data output from LIN-UART0	Input port	P13/TI04/TO04
LTxD1	Output	Serial data output from LIN-UART1	Input port	P10/TI00/$\overline{\text{SCK10}}$/CTxD/ TO00, P72/CTxD/KR2
LVIOUT	Output	Low-voltage detection flag output	Input port	P70/INTP5/KR0/TI15/ TO15
PCL	Output	Clock output	Output port	P140
<R> REGC	–	Connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF).	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
RESOUT	Output	RESET output	Output port	P130
RxD2	Input	Serial data input to UART2	Input port	P43/INTPR2/SDA20
$\overline{\text{SCK00}}$	I/O	Clock input/output for CSI00, CSI01, CSI10, and CSI11	Input port	P17/TI14/TO14, P60/SCL11
$\overline{\text{SCK01}}$				P76/KR6
$\overline{\text{SCK10}}$				P10/LTxD1/CTxD/TI00/ TO00
$\overline{\text{SCK11}}$				P153
SCL11	I/O	Clock input/output for simplified I^2C	Input port	P60/$\overline{\text{SCK00}}$
SCL20				P42/TxD2
SDA11	I/O	Serial data I/O for simplified I^2C	Input port	P61/SI00
SDA20				P43/RxD2/INTPR2
SI00	Input	Serial data input to CSI00, CSI01, CSI10, and CSI11	Input port	P16/TI12/TO12, P61/SDA11
SI01				P75/KR5
SI10				P11/LRxD1/CRxD/ INTPLR1/TI02/TO02
SI11				P152

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (3/5): 78K0R/HG3

Function Name	I/O	Function	After Reset	Alternate Function
SO00	Output	Serial data output from CSI00, CSI01, CSI10, and CSI11	Input port	P15/TI10/TO10, P62
SO01				P74/KR4
SO10				P12/INTP3/TI16/TO16
SO11				P151
SSI00	Input	Chip select input to CSI00, CSI01	Input port	P30/INTP2/TI01/TO01, P63
SSI01				P77/KR7
STOPST	Output	STOP status output	Input port	P31/INTP2/TI11/TO11, P52/TI22/TO22
TI00	Input	External count clock input to 16-bit timer 00	Input port	P10/SCK10/LTxD1/CTxD/TO00, P66/TO00
TI01		External count clock input to 16-bit timer 01		P30/SSI00/INTP2/TO01, P126/TO01
TI02		External count clock input to 16-bit timer 02		P11/SI10/LRxD1/INTPLR1/CRxD/TO02, P67/TO02
TI03		External count clock input to 16-bit timer 03		P125/INTP1/ADTRG/TO03, P127/TO03
TI04		External count clock input to 16-bit timer 04		P13/LTxD0/TO04, P01/TO04
TI05		External count clock input to 16-bit timer 05		P40/TOOL0/TO05, P00/INTP7/TO05
TI06		External count clock input to 16-bit timer 06		P14/LRxD0/INTPLR0/TO06, P02/TO06
TI07		External count clock input to 16-bit timer 07		P41/TOOL1/TO07, P44/TO07
TI10		External count clock input to 16-bit timer 10		P15/SO00/TO10, P45/TO10
TI11		External count clock input to 16-bit timer 11		P31/INTP2/STOPST/TO11, P54/TO11
TI12		External count clock input to 16-bit timer 12		P16/SI00/TO12, P46/TO12
TI13		External count clock input to 16-bit timer 13		P32/INTP4/TO13, P55/TO13
TI14		External count clock input to 16-bit timer 14		P17/SCK00/TO14, P64/TO14
TI15		External count clock input to 16-bit timer 15		P70/KR0/INTP5/TO15/LVIOUT, P56/TO15
TI16		External count clock input to 16-bit timer 16		P12/SO10/INTP3/TO16, P65/TO16
TI17		External count clock input to 16-bit timer 17		P71/KR1/INTP6/TO17, P57/TO17
TI20		External count clock input to 16-bit timer 20		P50/TO20/INTP3

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (4/5): 78K0R/HG3

Function Name	I/O	Function	After Reset	Alternate Function
TI21	Input	External count clock input to 16-bit timer 21	Input port	P51/TO21
TI22		External count clock input to 16-bit timer 22		P52/TO22/STOPST
TI23		External count clock input to 16-bit timer 23		P53/TO23
TI24		External count clock input to 16-bit timer 24		P154/TO24
TI25		External count clock input to 16-bit timer 25		P155/TO25
TI26		External count clock input to 16-bit timer 26		P156/TO26
TI27		External count clock input to 16-bit timer 27		P157/TO27
TO00	Output	16-bit timer 00 output	Input port	P10/SCK10/LTxD1/CTxD/TI00, P66/TI00
TO01		16-bit timer 01 output		P30/SSI00/INTP2/TI01, P126/TI01
TO02		16-bit timer 02 output		P11/SI10/LRxD1/INTPLR1/CRxD/TI02, P67/TI02
TO03		16-bit timer 03 output		P125/INTP1/ADTRG/TI03, P127/TI03
TO04		16-bit timer 04 output		P13/LTxD0/TI04, P01/TI04
TO05		16-bit timer 05 output		P40/TOOL0/TI05, P00/INTP7/TI05
TO06		16-bit timer 06 output		P14/LRxD0/INTPLR0/TI06, P02/TI06
TO07		16-bit timer 07 output		P41/TOOL1/TI07, P44/TI07
TO10		16-bit timer 10 output		P15/SO00/TI10, P45/TI10
TO11		16-bit timer 11 output		P31/INTP2/STOPST/TI11, P54/TI11
TO12		16-bit timer 12 output		P16/SI00/TI12, P46/TI12
TO13		16-bit timer 13 output		P32/INTP4/TI13, P55/TI13
TO14		16-bit timer 14 output		P17/SCK00/TI14, P64/TI14
TO15		16-bit timer 15 output		P70/KR0/INTP5/TI15/LVIOUT, P56/TI15
TO16		16-bit timer 16 output		P12/SO10/INTP3/TI16, P65/TI16
TO17		16-bit timer 17 output		P71/KR1/INTP6/TI17, P57/TI17
TO20		16-bit timer 20 output		P50/TI20/INTP3

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(2) Non-port functions (5/5): 78K0R/HG3

Function Name	I/O	Function	After Reset	Alternate Function
TO21	Output	16-bit timer 21 output	Input port	P51/TI21
TO22		16-bit timer 22 output		P52/TI22/STOPST
TO23		16-bit timer 23 output		P53/TI23
TO24		16-bit timer 24 output		P154/TI24
TO25		16-bit timer 25 output		P155/TI25
TO26		16-bit timer 26 output		P156/TI26
TO27		16-bit timer 27 output		P157/TI27
TOOL0	I/O	Data I/O for flash memory programmer/debugger	Input port	P40/TI05/TO05
TOOL1	Output	Clock output for debugger	Input port	P41/TI07/TO07
TxD2	Output	Serial data output from UART2	Input port	P42/SCL20
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
EXCLKS	Input	External clock input for subclock	Input port	P124
V_{DD}	–	Positive power supply (P121 to P124 and other than ports (other than $\overline{\text{RESET}}$ pin and FLMD0 pin))	–	–
EV_{DD0}, EV_{DD1}	–	Positive power supply for ports (other than P80 to P87, P90 to P97, P100 to P107, and P121 to P124), and $\overline{\text{RESET}}$ and FLMD0 pin. Make EV_{DD0} and EV_{DD1} the same potential as V_{DD}.	–	–
AV_{REF}	–	• A/D converter and comparator reference voltage input • Positive power supply for P80 to P87, P90 to P97, P100 to P107, and A/D converter	–	–
V_{SS}	–	Ground potential (P121 to P124 and other than ports (other than $\overline{\text{RESET}}$ pin and FLMD0 pin))	–	–
EV_{SS0}, EV_{SS1}	–	Ground potential for ports (other than P80 to P87, P90 to P97, P100 to P107, and P121 to P124), and $\overline{\text{RESET}}$ and FLMD0 pin. Make EV_{SS0} and EV_{SS1} the same potential as V_{SS}.	–	–
AV_{SS}	–	Ground potential for A/D converter, P80 to P87, P90 to P97, and P100 to P107. Make AV_{SS} the same potential as V_{SS}, EV_{SS0} and EV_{SS1}.	–	–

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

2.2 Description of Pin Functions

Remark The pins mounted depend on the product. See **1.4 Ordering Information (Top View)** and **2.1 Pin Function List**.

2.2.1 P00 to P03 (port 0)

P00 to P03 function as an I/O port. P00 to P03 pins also function as timer I/O, and external interrupt request input.

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P00/INTP7/TI05/TO05	√	√	√	√
P01/TI04/TO04	–	–	√	√
P02/TI06/TO06	–	–	√	√
P03	–	–	–	√

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

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

(1) Port mode

P00 to P03 function as an I/O port. P00 to P03 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 to P03 function as timer I/O, and external interrupt request input.

(a) INTP7

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

(b) TI04 to TI06

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 04 to 06.

(c) TO04 to TO06

These are the timer output pins of 16-bit timers 04 to 06.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

2.2.2 P10 to P17 (port 1)

<R> P10 to P17 function as an I/O port. These pins also function as external interrupt request input, serial interface data I/O, clock I/O, and timer I/O.

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P10//TI00/ SCK10/LTxD1/ CTxD/TO00	√			
P11/TI02/ SI10/LRxD1/CRxD/ INTPLR1/TO02	√			
P12/INTP3/TI16/ SO10/TO16	√			
P13/TI04/LTxD0/ TO04	√			
P14/TI06/LRxD0/ INTPLR0/TO06	√			
P15/TI10/SO00/ TO10	√			
P16/TI12/SI00/ TO12	√			
P17/TI14/SCK00/ TO14	√			

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted

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

(1) Port mode

P10 to P17 function as an I/O port. P10 to P17 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 P17 function as external interrupt request input, serial interface data I/O, clock I/O, and timer I/O.

(a) INTP3

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) SI00, SI10

These are the serial data input pins of serial interface CSI00, CSI10.

(c) CRxD

This is a CAN data input pin of CAN.

(d) CTxD

This is a CAN data output pin of CAN.

(e) SO00, SO10

These are the serial data output pins of serial interface CSI00, CSI10.

(f) $\overline{\text{SCK00}}$, $\overline{\text{SCK10}}$

These are the serial clock I/O pins of serial interface CSI00, CSI10.

(g) LRxD0, LRxD1

These are the serial data input pins of serial interface LIN-UART0, LIN-UART1.

(h) LTxD0, LTxD1

These are the serial data output pins of serial interface LIN-UART0, LIN-UART1.

(i) INTPLR0, INTPLR1

These are the external interrupt request input pins for which the valid edge for LIN-UART0, LIN-UART1 (rising edge, falling edge, or both rising and falling edges) can be specified.

(j) TI00, TI02, TI04, TI06, TI10, TI12, TI14, TI16

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 00, 02, 04, 06, 10, 12, 14, and 16.

(k) TO00, TO02, TO04, TO06, TO10, TO12, TO14, TO16

These are the timer output pins of 16-bit timers 00, 02, 04, 06, 10, 12, 14, and 16.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

2.2.3 P30 to P32 (port 3)

P30 to P32 function as an I/O port. P30 to P32 pins also function as external interrupt request input, serial interface chip select input, timer I/O, and STOP status output.

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P30/INTP2/SSI00/TI01/TO01	√			
P31/INTP2/TI11/STOPST/TO11	√			
P32/INTP4/TI13/TO13	√			

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

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

(1) Port mode

P30 to P32 function as an I/O port. P30 to P32 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

P30 to P32 function as external interrupt request input, serial interface chip select input, timer I/O, and STOP status output.

(a) INTP2, INTP4

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) SSI00

This is a chip select input pin of serial interface CSI00.

(c) STOPST

This is a STOP status output pin.

(d) TI01, TI11, TI13

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 01, 11, and 13.

(e) TO01, TO11, TO13

These are the timer output pins of 16-bit timers 01, 11, and 13.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

2.2.4 P40 to P47 (port 4)

P40 to P47 function as an I/O port. These pins also function as external interrupt request input, serial interface data I/O, clock I/O, data I/O for a flash memory programmer/debugger, and timer I/O.

Output from the P42 and P43 pins can be specified as normal CMOS output or N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 4 (POM4).

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P40/TOOL0/TI05/TO05	√	√	√	√
P41/TOOL1/TI07/TO07	√	√	√	√
P42/TxD2/SCL20	–	√	√	√
P43/RxD2/SDA20/INTPR2	–	√	√	√
P44/TI07/TO07	–	–	√	√
P45/TI10/TO10	–	–	√	√
P46/TI12/TO12	–	–	√	√
P47/INTP8	–	–	√	√

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

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

(1) Port mode

P40 to P47 function as an I/O port. P40 to P47 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 to P47 function as external interrupt request input, serial interface data I/O, clock I/O, data I/O for a flash memory programmer/debugger, and timer I/O.

(a) INTP8

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.

(b) TI05, TI07, TI10, TI12

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 05, 07, 10, and 12.

(c) TO05, TO07, TO10, TO12

These are the timer output pins from 16-bit timers 05, 07, 10, and 12.

(d) 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).

(e) 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).

(f) RxD2

This is a serial data input pin of serial interface UART2.

(g) TxD2

This is a serial data output pin of serial interface UART2.

(h) SDA11

This is a serial data I/O pin of serial interface for simplified I^2C.

(i) SCL11

This is a serial clock I/O pin of serial interface for simplified I^2C.

(j) INTPR2

This is the external interrupt request input pin for which the valid edge for UART2 (rising edge, falling edge, or both rising and falling edges) can be specified.

Cautions 1. The shaded pins are provided at two ports. Select either port by using the corresponding register.

2. 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 EV_{DD0} or EV_{DD1} 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 EV_{DD0} or EV_{DD1} via an external resistor.

2.2.5 P50 to P57 (port 5)

P50 to P57 function as an I/O port. These pins also function as external interrupt request input, timer I/O, and STOP status output.

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P50/INTP3/TI20/TO20	–	√	√	√
P51/TI21/TO21	–	√	√	√
P52/TI22/STOPST/TO22	–	√	√	√
P53/TI23/TO23	–	√	√	√
P54/TI11/TO11	–	–	√	√
P55/TI13/TO13	–	–	√	√
P56/TI15/TO15	–	–	√	√
P57/TI17/TO17	–	–	√	√

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

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 external interrupt request input, timer I/O, and STOP status output.

(a) 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.

(b) TI11, TI13, TI15, TI17, TI20 to TI23

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 11, 13, 15, 17, and 20 to 23.

(c) TO11, TO13, TO15, TO17, TO20 to TO23

These are the timer output pins of 16-bit timers 11, 13, 15, 17, and 20 to 23.

(d) STOPST

This is a STOP status output pin.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

2.2.6 P60 to P67 (port 6)

P60 to P67 function as an I/O port. P60 to P67 pins also function as serial interface data I/O, clock I/O, chip select input, and timer I/O.

Input to the P60, P61 and P63 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units, using port input mode register 6 (PIM6).

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P60/SCK00/SCL11	√	√	√	√
P61/SI00/SDA11	√	√	√	√
P62/SO00	√	√	√	√
P63/SSI00	√	√	√	√
P64/TI14/TO14	–	–	√	√
P65/TI16/TO16	–	–	√	√
P66/TI00/TO00	–	–	√	√
P67/TI02/TO02	–	–	√	√

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

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

(1) Port mode

P60 to P67 function as an I/O port. P60 to P67 can be set to input port or output port in 1-bit units using port mode register 6 (PM6). P60 to P67, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 6 (PU6).

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

(2) Control mode

P60 to P67 function as serial interface data I/O, clock I/O, chip select input, and timer I/O.

(a) TI00, TI02, TI14, TI16

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 00, 02, 14, and 16.

(b) TO00, TO02, TO14, TO16

These are the timer output pins of 16-bit timers 00, 02, 14, and 16.

(c) $\overline{\text{SCK00}}$

This is a serial clock I/O pin of serial interface CSI00.

(d) SI00

This is a serial data input pin of serial interface CSI00.

(e) SO00

This is a serial data output pin of serial interface CSI00.

(f) $\overline{\text{SSI00}}$

This is a chip select input pin of serial interface CSI00.

(g) SDA11

This is a serial data I/O pin of serial interface for simplified I^2C.

(h) SCL11

This is a serial clock I/O pin of serial interface for simplified I^2C.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

2.2.7 P70 to P77 (port 7)

P70 to P77 function as an I/O port. These pins also function as external interrupt request input, key interrupt input, CAN data I/O, serial interface data I/O, clock I/O, timer I/O, and low-voltage detection flag output.

Input to the P73, P75 to P77 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units, using port input mode register 7 (PIM7).

Output from the P72, P74, and P76 pins can be specified as normal CMOS output or N-ch open-drain output (V_{DD} tolerance) in 1-bit units, using port output mode register 7 (POM7).

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P70/INTP5/KR0/TI15/TO15/LVIOUT	√	√	√	√
P71/INTP6/KR1/TI17/TO17	√	√	√	√
P72/KR2/CTxD/LTxD1	√	√	√	√
P73/KR3/CRxD/LRxD1/INTPLR1	√	√	√	√
P74/KR4/SO01	–	√	√	√
P75/KR5/SI01	–	√	√	√
P76/KR6/$\overline{\text{SCK01}}$	–	√	√	√
P77/KR7/$\overline{\text{SSI01}}$	–	√	√	√

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

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

(1) Port mode

P70 to P77 function as an I/O port. P70 to P77 can be set to input or output port in 1-bit units using port mode register 7 (PM7). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

(2) Control mode

P70 to P77 function as external interrupt request input, key interrupt input, CAN data I/O, serial interface data I/O, clock I/O, timer I/O, and low-voltage detection flag output.

(a) INTP5, INTP6

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) KR0 to KR7

These are the key interrupt input pins

(c) CRxD

This is a serial data input pin of CAN.

(d) CTxD

This is a serial data output pin of CAN.

(e) LRxD1

This is a serial data input pin of serial interface LIN-UART1.

(f) LTxD1

This is a serial data output pin of serial interface LIN-UART1.

(g) INTPLR1

This is a external interrupt request input pin for which the valid edge for LIN-UART1 (rising edge, falling edge, or both rising and falling edges) can be specified.

(h) $\overline{\text{SCK01}}$

This is a serial clock I/O pin of serial interface CSI01.

(i) SI01

This is a serial data input pin of serial interface CSI01.

(j) SO01

This is a serial data output pin of serial interface CSI01.

(k) $\overline{\text{SSI01}}$

This is a chip select input pin of serial interface CSI01.

(l) TI15, TI17

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 15 and 17.

(m) TO15, TO17

These are the timer output pins of 16-bit timers 15 and 17.

(n) LVIOUT

This is a low-voltage detection flag output pin.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

2.2.8 P80 to P87 (port 8)

P80 to P87 function as an I/O port. These pins also function as A/D converter analog input.

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P80/ANI00	√			
P81/ANI01	√			
P82/ANI02	√			
P83/ANI03	√			
P84/ANI04	√			
P85/ANI05	√			
P86/ANI06	√			
P87/ANI07	√			

Remark √: Mounted

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

(1) Port mode

P80 to P87 function as an I/O port. P80 to P87 can be set to input or output port in 1-bit units using port mode register 8 (PM8).

(2) Control mode

P80 to P87 function as A/D converter analog input pins (ANI00 to ANI07). When using these pins as analog input pins, see **10.6 (6) ANI00/P80 to ANI07/P87 and ANI08/P90 to ANI15/P97 and ANI16/P100 to ANI23/P107**.

Caution ANI00/P80 to ANI07/P87 are set in the digital input (general-purpose port) mode after release of reset.

2.2.9 P90 to P97 (port 9)

P90 to P97 function as an I/O port. These pins also function as A/D converter analog input.

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P90/ANI08	√	√	√	√
P91/ANI09	√	√	√	√
P92/ANI10	√	√	√	√
P93/ANI11	–	√	√	√
P94/ANI12	–	√	√	√
P95/ANI13	–	√	√	√
P96/ANI14	–	√	√	√
P97/ANI15	–	–	√	√

Remark √: Mounted, –: Not mounted

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).

(2) Control mode

P90 to P97 function as A/D converter analog input pins (ANI08 to ANI15). When using these pins as analog input pins, see **10.6 (6) ANI00/P80 to ANI07/P87 and ANI08/P90 to ANI15/P97 and ANI16/P100 to ANI23/P107**.

Caution ANI08/P90 to ANI15/P97 are set in the digital input (general-purpose port) mode after release of reset.

2.2.10 P100 to P107 (port 10)

P100 to P107 function as an I/O port. These pins also function as A/D converter analog input.

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P100/ANI16	–	–	–	√
P101/ANI17	–	–	–	√
P102/ANI18	–	–	–	√
P103/ANI19	–	–	–	√
P104/ANI20	–	–	–	√
P105/ANI21	–	–	–	√
P106/ANI22	–	–	–	√
P107/ANI23	–	–	–	√

Remark √: Mounted, –: Not mounted

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

(1) Port mode

P100 to P107 function as an I/O port. P100 to P107 can be set to input or output port in 1-bit units using port mode register 10 (PM10).

(2) Control mode

P100 to P107 function as A/D converter analog input pins (ANI16 to ANI23). When using these pins as analog input pins, see **10.6 (6) ANI00/P80 to ANI07/P87 and ANI08/P90 to ANI15/P97 and ANI16/P100 to ANI23/P107**.

Caution ANI16/P100 to ANI23/P107 are set in the digital input (general-purpose port) mode after release of reset.

2.2.11 P120 to P127 (port 12)

P120 and P125 toP127 function as a 4-bit I/O port. P121 to P124 function as a 4-bit input port. These pins also function as external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, external clock input for main system clock, external clock input for subclock, external trigger input for A/D converter, and timer I/O.

	78K0R/HC3 (µPD78F10yy)	78K0R/HE3 (µPD78F10yy)	78K0R/HF3 (µPD78F10yy)	78K0R/HG3 (µPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P120/INTP0/EXLVI/ TI11/TO11	√	√	**Note**	**Note**
P121/X1	√	√	√	√
P122/X2/EXCLK	√	√	√	√
P123	√	√	√	√
P124/EXCLKS	√	√	√	√
P125/INTP1/ ADTRG/TI03/TO03	√	√	√	√
P126/TI01/TO01	–	–	√	√
P127/TI03/TO03	–	–	–	√

Note TI11, TO11 pins are not mounted. Port functions other than TI11 and TO11 as well as shared functions are provided.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

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

(1) Port mode

P120 and P125 to P127 function as a 4-bit I/O port. P120 and P125 to P127 can be set to input 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 to P124 function as a 4-bit input port.

(2) Control mode

P120 to P127 function as external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, external clock input for main system clock, external clock input for subclock, external trigger input for A/D converter, and timer I/O.

(a) INTP0, INTP1

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) EXLVI

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

(c) X1, X2

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

(d) EXCLK

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

(e) EXCLKS

This is an external clock input pin for subclock.

(f) ADTRG

This is an external trigger input pin for A/D converter

(g) TI01, TI03, TI11

These are the pins for inputting an external count clock/capture trigger to 16-bit timer 01, 03, and 11.

(h) TO01, TO03, TO11

These are the timer output pins from 16-bit timer 01, 03, and 11.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

2.2.12 P130 (port 13)

P130 functions as a 1-bit output port.

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P130/RESOUT	√	√	√	√

Remarks 1. When the device is reset, P130 outputs a low level.

2. √: Mounted, –: Not mounted

(1) Port mode

P130 functions as a 1-bit output port.

(2) Control mode

P130 functions as reset output.

(a) RESOUT

This is a pin for reset output.

2.2.13 P140 (port 14)

P140 functions as an I/O port. P140 pin also functions as clock output.

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P140/PCL	√	√	√	√

Remarks 1. When the device is reset, P140 outputs a low level.

2. √: Mounted, –: Not mounted

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

(1) Port mode

P140 functions as an I/O port. P140 can be set to input or output port in 1-bit unit 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 functions as clock output.

(a) PCL

This is a clock output pin.

2.2.14 P150 to P157 (port 15)

P150 to P157 function as an I/O port. These pins also function as serial interface data I/O, clock I/O, and timer I/O.

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P150	–	–	–	√
P151/SO11	–	–	–	√
P152/SI11	–	–	–	√
P153/$\overline{\text{SCK11}}$	–	–	–	√
P154/TI24/TO24	–	–	–	√
P155/TI25/TO25	–	–	–	√
P156/TI26/TO26	–	–	–	√
P157/TI27/TO27	–	–	–	√

Remark √: Mounted, –: Not mounted

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

(1) Port mode

P150 to P157 function as an I/O port. P150 to P157 can be set to input or output port in 1-bit units using port mode register 15 (PM15).

(2) Control mode

P150 to P157 function as serial interface data I/O, clock I/O, and timer I/O.

(a) $\overline{\text{SCK11}}$

This is a serial clock I/O pin of serial interface CSI11.

(b) SI11

This is a serial data input pin of serial interface CSI11.

(c) SO11

This is a serial data output pin of serial interface CSI11.

(d) TI24 to TI27

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 24 to 27.

(e) TO24 to TO27

These are the timer output pins of 16-bit timers 24 to 27.

2.2.15 AV$_{\text{REF}}$

This is the A/D converter reference voltage input pin and the positive power supply pin of P80 to P87, P90 to P97, P100 to P107, and A/D converter.

When all pins of ports 8 to 10 are used as the analog port pins, make the potential of AV$_{\text{REF}}$ be such that 2.7 V $\leq$ AV$_{\text{REF}}$ $\leq$ V$_{\text{DD}}$. When one or more of the pins of ports 8 to 10 are used as the digital port pins or when the A/D converter is not used, make AV$_{\text{REF}}$ the same potential as EV$_{\text{DD}}$, EV$_{\text{DD0}}$, EV$_{\text{DD1}}$, or V$_{\text{DD}}$.

2.2.16 AV$_{SS}$

This is the ground potential pin of A/D converter, P80 to P87, P90 to P97, and P100 to P107. Even when the A/D converter is not used, always use this pin with the same potential as EV$_{SS}$, EV$_{SS0}$, EV$_{SS1}$, and V$_{SS}$.

2.2.17 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

When the external reset pin is not used, connect this pin directly or via a resistor to EV$_{DD}$[Note].

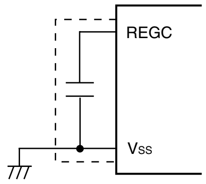
When the external reset pin is used, design the circuit based on V$_{DD}$.

Note EV$_{DD0}$ and EV$_{DD1}$ in the 78K0R/HG3.

2.2.18 REGC

This is the pin for connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect this pin
<R> 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.

2.2.19 V$_{DD}$, EV$_{DD}$, EV$_{DD0}$, EV$_{DD1}$

V$_{DD}$ is the positive power supply pin for P121 to P124 and pins other than ports (excluding the $\overline{\text{RESET}}$ and FLMD0 pins).

EV$_{DD}$, EV$_{DD0}$, and EV$_{DD1}$ are the positive power supply pins for ports other than P80 to P87, P90 to P97, P100 to P107, and P121 to P124 as well as for the $\overline{\text{RESET}}$ and FLMD0 pins.

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
V$_{DD}$	√	√	√	√
EV$_{DD}$	√	√	√	–
EV$_{DD0}$	–	–	–	√
EV$_{DD1}$	–	–	–	√

Remark √: Mounted, –: Not mounted

2.2.20 V_{SS}, EV_{SS}, EV_{SS0}, EV_{SS1}

V_{SS} is the ground potential pin for P121 to P124 and pins other than ports (excluding the $\overline{\text{RESET}}$ and FLMD0 pins).

EV_{SS}, EV_{SS0}, and EV_{SS1} are the ground potential pins for ports other than P80 to P87, P90 to P97, P100 to P107, and P121 to P124 as well as for the $\overline{\text{RESET}}$ and FLMD0 pins.

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
V_{SS}	√	√	√	√
EV_{SS}	√	√	√	–
EV_{SS0}	–	–	–	√
EV_{SS1}	–	–	–	√

Remark √: Mounted, –: Not mounted

2.2.21 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
<R> **24.2 (2) Back ground event control register**). To pull it down externally, use a resistor of 100 kΩ or more.

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,
<R> use a resistor of 100 kΩ or more.

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
<R> it down externally, use a resistor of 100 kΩ or more.

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

Table 2-3 shows the types of pin I/O circuits and the recommended connections of unused pins.

Remark The pins mounted depend on the product. See **1.4 Ordering Information (Top View)** and **2.1 Pin Function List**.

Table 2-3. Connection of Unused Pins (1/4)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/INTP7/TI05/TO05	8-R	I/O	Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P01/TI04/TO04			
P02/TI06/TO06			
P03			
P10/TI00/$\overline{SCK10}$/TO00/LTxD1/CTxD	8-T		
P11/TI02/SI10/LRxD1/CRxD/INTPLR1/TO02	8-R		
P12/INTP3/TI16/SO10/TO16	8-T		
P13/TI04/LTxD0/TO04	8-R		
P14/TI06/LRxD0/INTPLR0/TO06			
P15/TI10/SO00/TO10			
P16/TI12/SI00/TO12			
P17/TI14/SCK00/TO14			
P30/INTP2/$\overline{SSI00}$/TI01/TO01	8-T		
P31/INTP2/TI11/STOPST/TO11	8-R		
P32/INTP4/TI13/TO13			
P40/TOOL0/TI05/TO05			<When on-chip debugging is enabled> Pull this pin up (pulling it down is prohibited). Recommended pull-up 10 kΩ[Note]. <When on-chip debugging is disabled> Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P41/TOOL1/TI07/TO07			Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P42/TxD2/SCL20	5-AR		Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level output.
P43/RxD2/SDA20/INTPR2			

Note In case on-chip debugging is enabled the external pull-up resistor is mandatory to ensure a proper operation of the device when the QB-MINI2 is not connected.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Table 2-3. Connection of Unused Pins (2/4)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P44/TI07/TO07	8-R	I/O	Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P45/TI10/TO10			
P46/TI12/TO12			
P47/INTP8			
P50/INTP3/TI20/TO20			
P51/TI21/TO21			
P52/TI22/STOPST/TO22			
P53/TI23/TO23			
P54/TI11/TO11			
P55/TI13/TO13			
P56/TI15/TO15			
P57/TI17/TO17			
P60/SCK00/SCL11	13-AK		Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor, or connect directly to EV_{SS}, EV_{SS0}, or EV_{SS1}. Output: Leave these pins open by setting the port output latches to 0 and outputting low levels, or independently connect them to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor by setting the port output latches to 1.
P61/SI00/SDA11			
P62/SO00	13-AL		
P63/SSI00	13-AK		
P64/TI14/TO14	8-R		Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P65/TI16/TO16			
P66/TI00/TO00			
P67/TI02/TO02			
P70/INTP5/KR0/TI15/TO15/LVIOUT			
P71/INTP6/KR1/TI17/TO17			
P72/KR2/CTxD/LTxD1	5-AR		Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level output.
P73/KR3/CRxD/LRxD1/INTPLR1	5-AN		Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Table 2-3. Connection of Unused Pins (3/4)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P74/KR4/SO01	5-AZ	I/O	Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level output.
P75/KR5/SI01	5-AN		Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P76/KR6/$\overline{\text{SCK01}}$	5-AY		Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level output.
P77/KR7/$\overline{\text{SSI01}}$	5-AN		Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P80/ANI00 to P87/ANI07[Note 1]	11-G		Input: Independently connect to AV_{REF} or AV_{SS} via a resistor. Output: Leave open.
P90/ANI08 to P97/ANI15[Note 1]			
P100/ANI16 to P107/ANI23[Note 1]			
P120/INTP0/EXLVI/TI11/TO11	8-R		Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P121/X1[Note 2]	37-E	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK[Note 2]			
P123[Note 2]	2-H		
P124/EXCLKS[Note 2]	37-F		
P125/INTP1/ADTRG/TI03/TO03	8-R	I/O	Input: Independently connect to AV_{REF} or AV_{SS} via a resistor. Output: Leave open.
P126/TI01/TO01			
P127/TI03/TO03			

Notes 1. P80/ANI00 to P87/ANI07, P90/ANI08 to P97/ANI15, and P100/ANI16 to P107/ANI23 are set in the digital input port mode after release of reset.

2. Use recommended connection above in input port mode (see **Figure 5-3 Format of Clock Operation Mode Control Register (CMC)**) when these pins are not used.

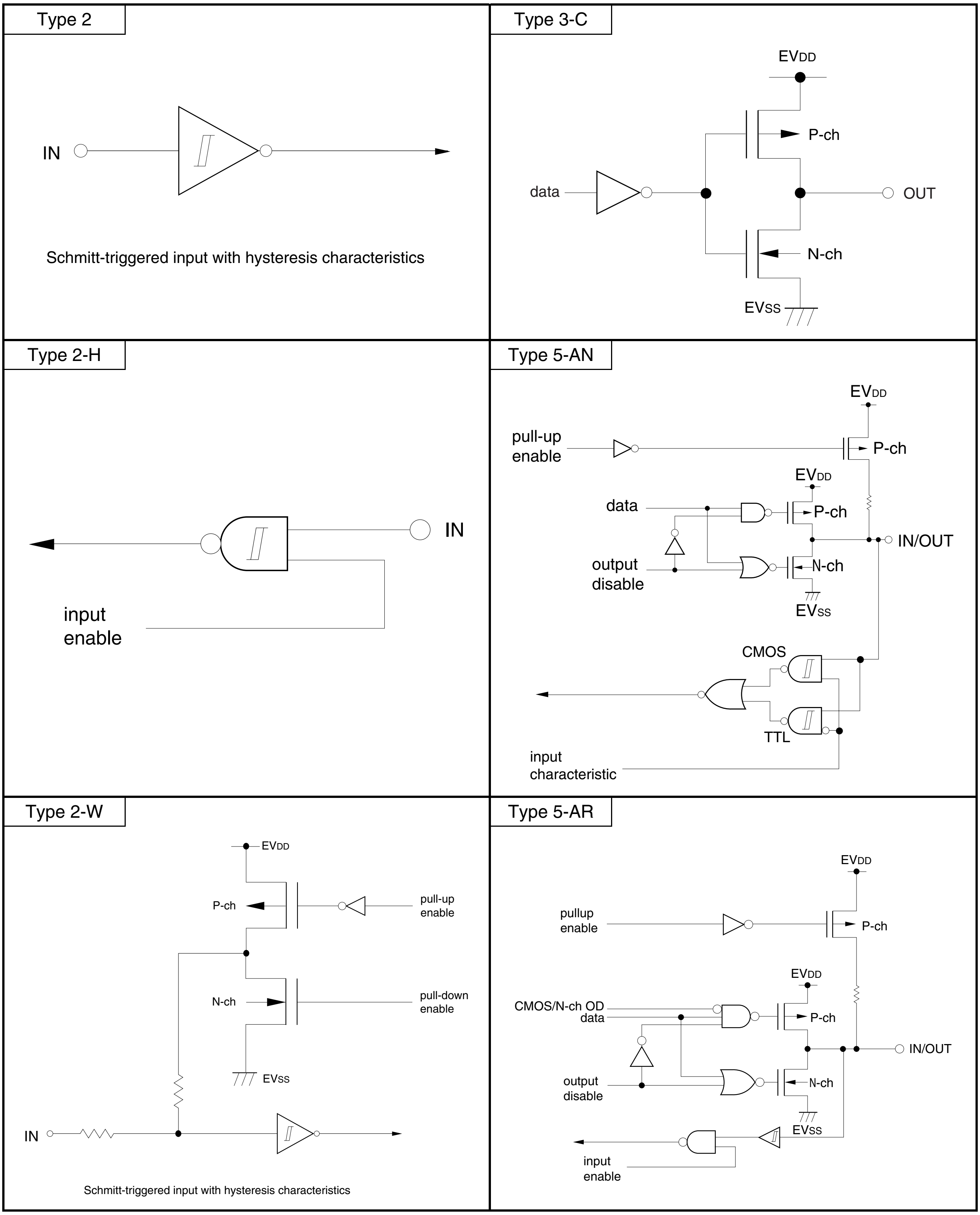
Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Table 2-3. Connection of Unused Pins (4/4)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P130/RESOUT	3-C	Output	Leave open.
P140/PCL	8-T	I/O	Input: Independently connect to EV_{DD}, EV_{DD0}, EV_{DD1}, EV_{SS}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P150	8-R		
P151/SO11			
P152/SI11			
P153/$\overline{\text{SCK11}}$			
P154/TI24/TO24			
P155/TI25/TO25			
P156/TI26/TO26			
P157/TI27/TO27			
<R> AV_{REF}	–	–	<When one or more of P80 to P87 and P90 to P97 and P100 to P107 are set as a digital port> Make this pin the same potential as EV_{DD}, EV_{DD0}, EV_{DD1}, or V_{DD}. <When all of P80 to P87 and P90 to P97 and P100 to P107 are set as analog ports> Make this pin to have a potential where 2.7 V ≤ AV_{REF} ≤ V_{DD}. In addition, when the following conditions are fulfilled, it is possible to reduce current consumption current during AV_{REF} = 0 V. • It is ADCS=0 and ADCE=0. • Positive supply voltage should not be applied to all the analog ports (ANI0-ANI23). • Positive supply voltage should not be applied to all the digital ports (P80-P87, P90-P97, P100-P107). • All the digital ports (P80-P87, P90-P97, P100-P107) have not carried out "H" output.
AV_{SS}	–	–	Make this pin the same potential as the EV_{SS}, EV_{SS0}, EV_{SS1}, or V_{SS}.
FLMD0	2-W	–	Leave open or connect to V_{SS} via a resistor of 100 kΩ or more.
$\overline{\text{RESET}}$	2	Input	Connect directly to V_{DD} or via a resistor.
<R> REGC	–	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).

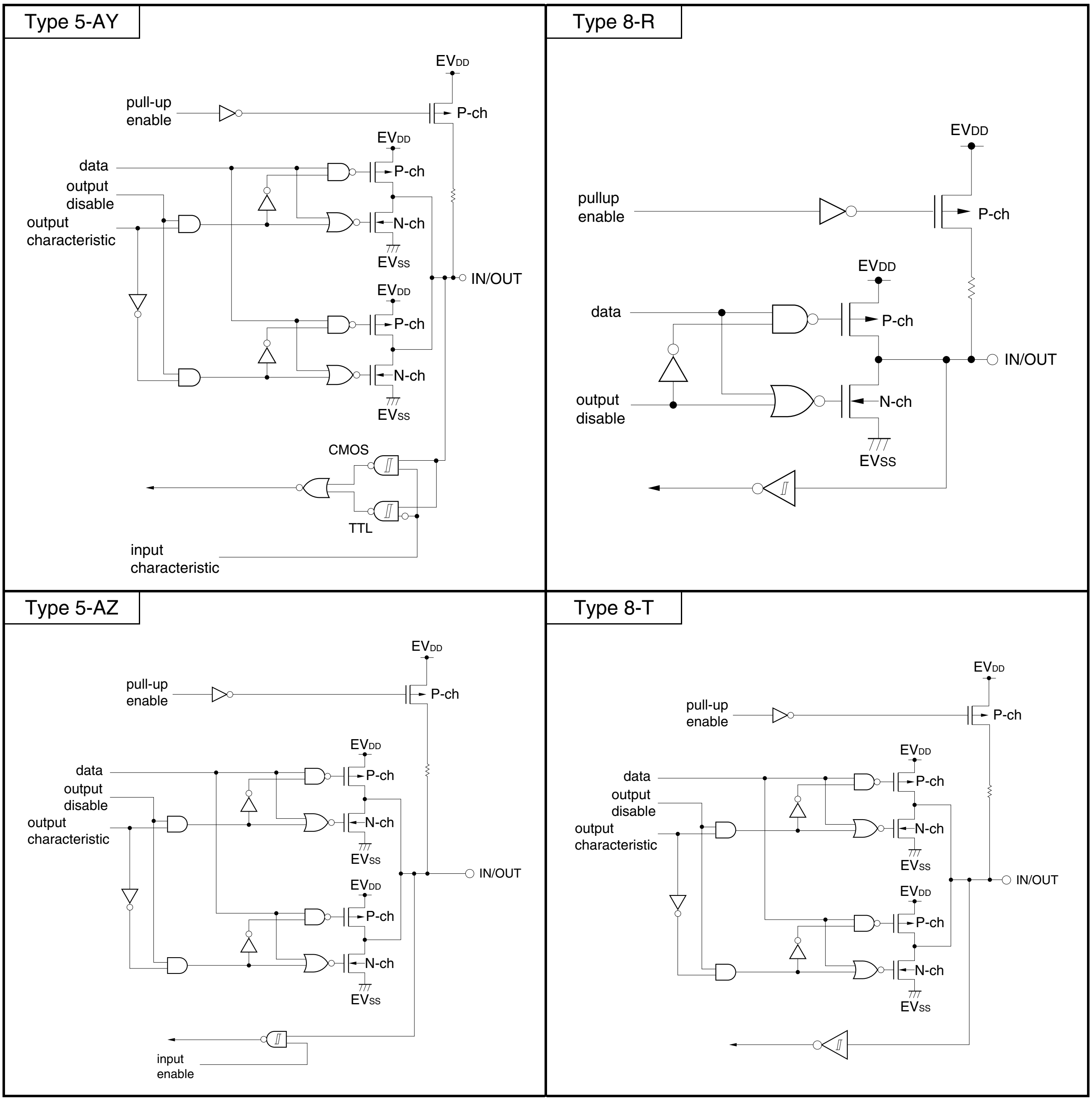
Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

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



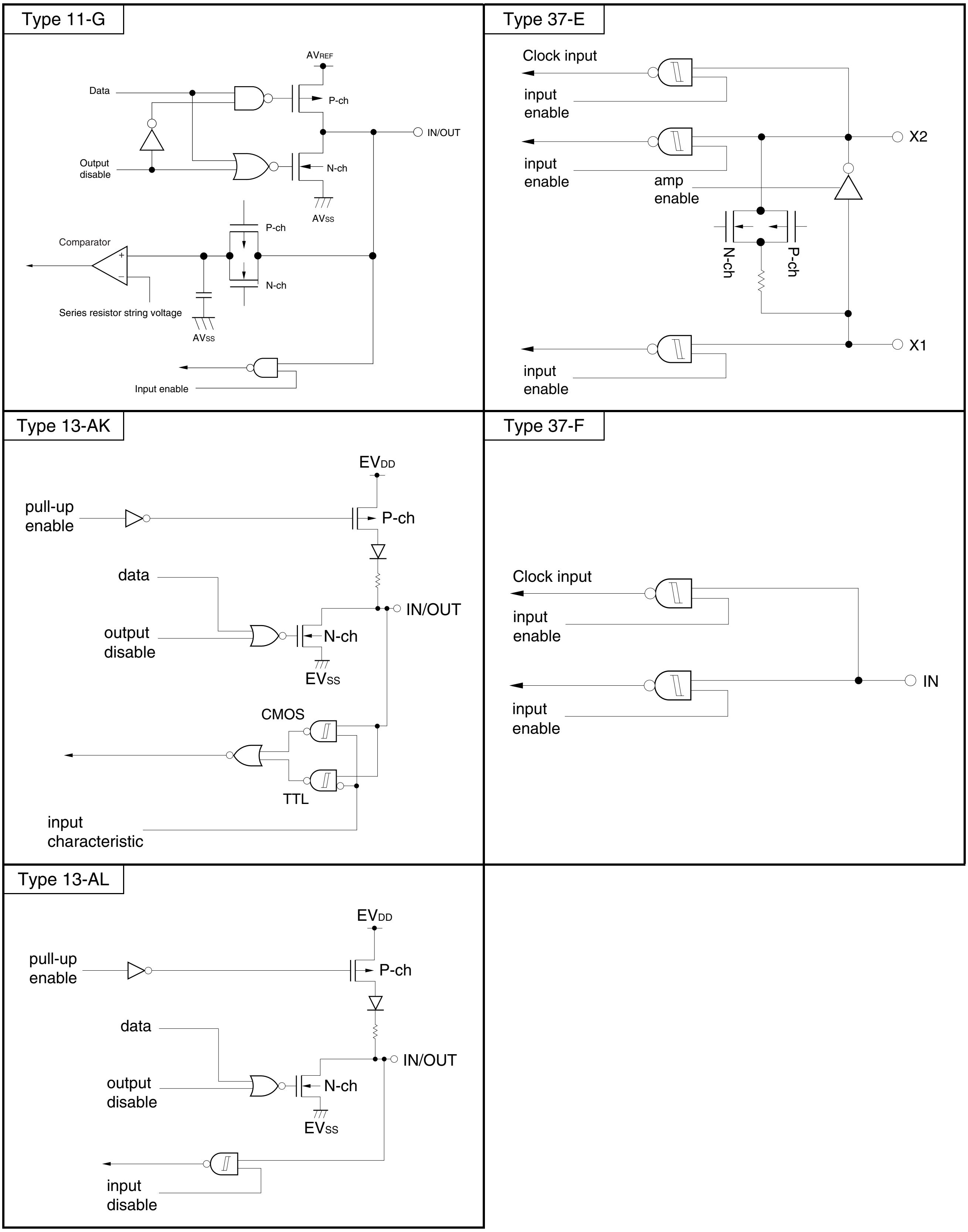
Remark When using the 78K0R/HG3, read EV_{DD} as EV_{DD0} and EV_{DD1}, and EV_{SS} as EV_{SS0} and EV_{SS1}.

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



Remark When using the 78K0R/HG3, read EV_{DD} as EV_{DD0} and EV_{DD1}, and EV_{SS} as EV_{SS0} and EV_{SS1}.

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



Type 11-G

Type 37-E

Type 13-AK

Type 37-F

Type 13-AL

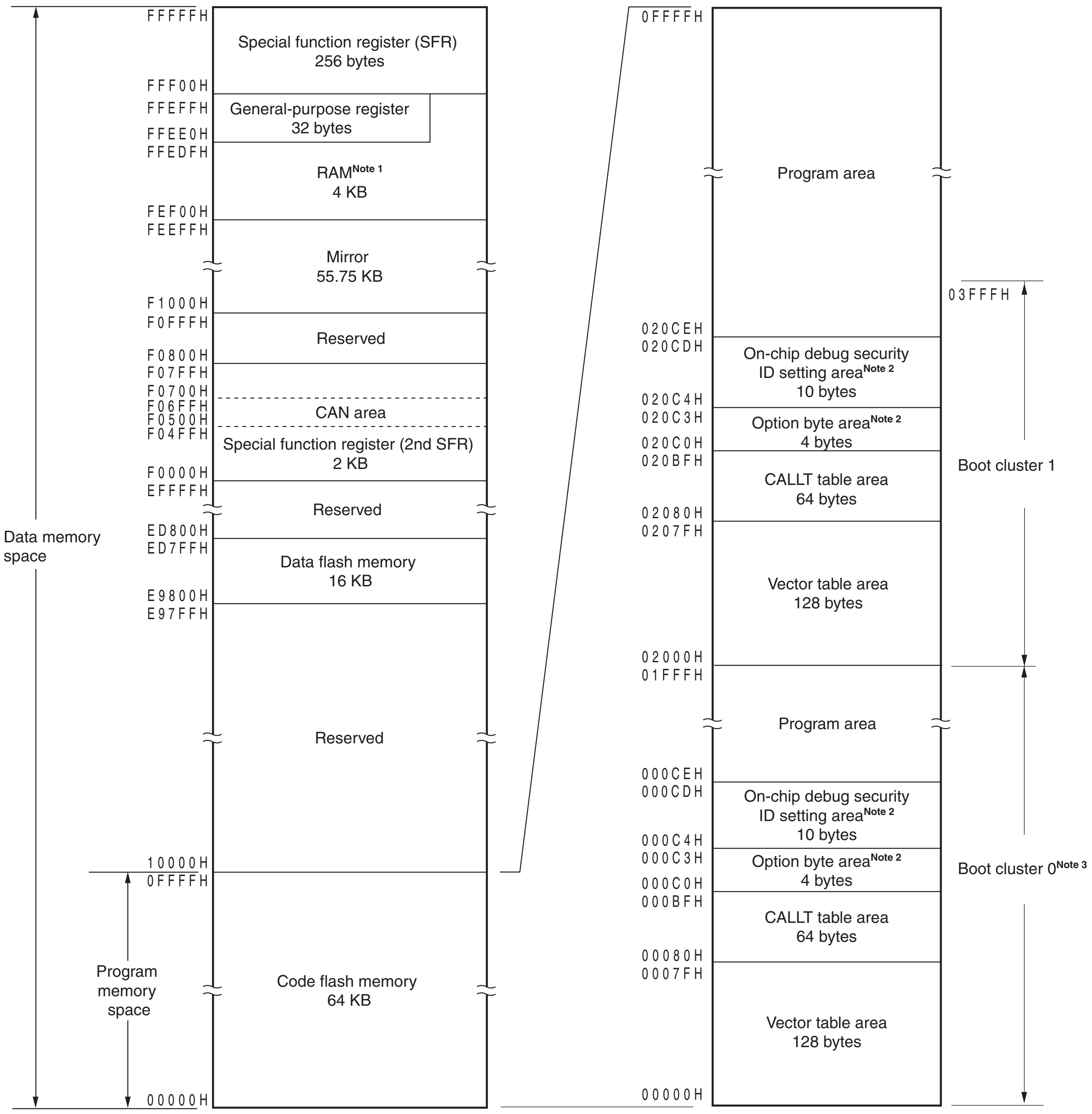
Remark When using the 78K0R/HG3, read EV_{DD} as EV_{DD0} and EV_{DD1}, and EV_{SS} as EV_{SS0} and EV_{SS1}.

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the 78K0R/Hx3 can access a 1 MB memory space. Figures 3-1 to 3-5 show the memory maps.

Figure 3-1. Memory Map (μPD78F1031, 78F1036, 78F1041, 78F1046)



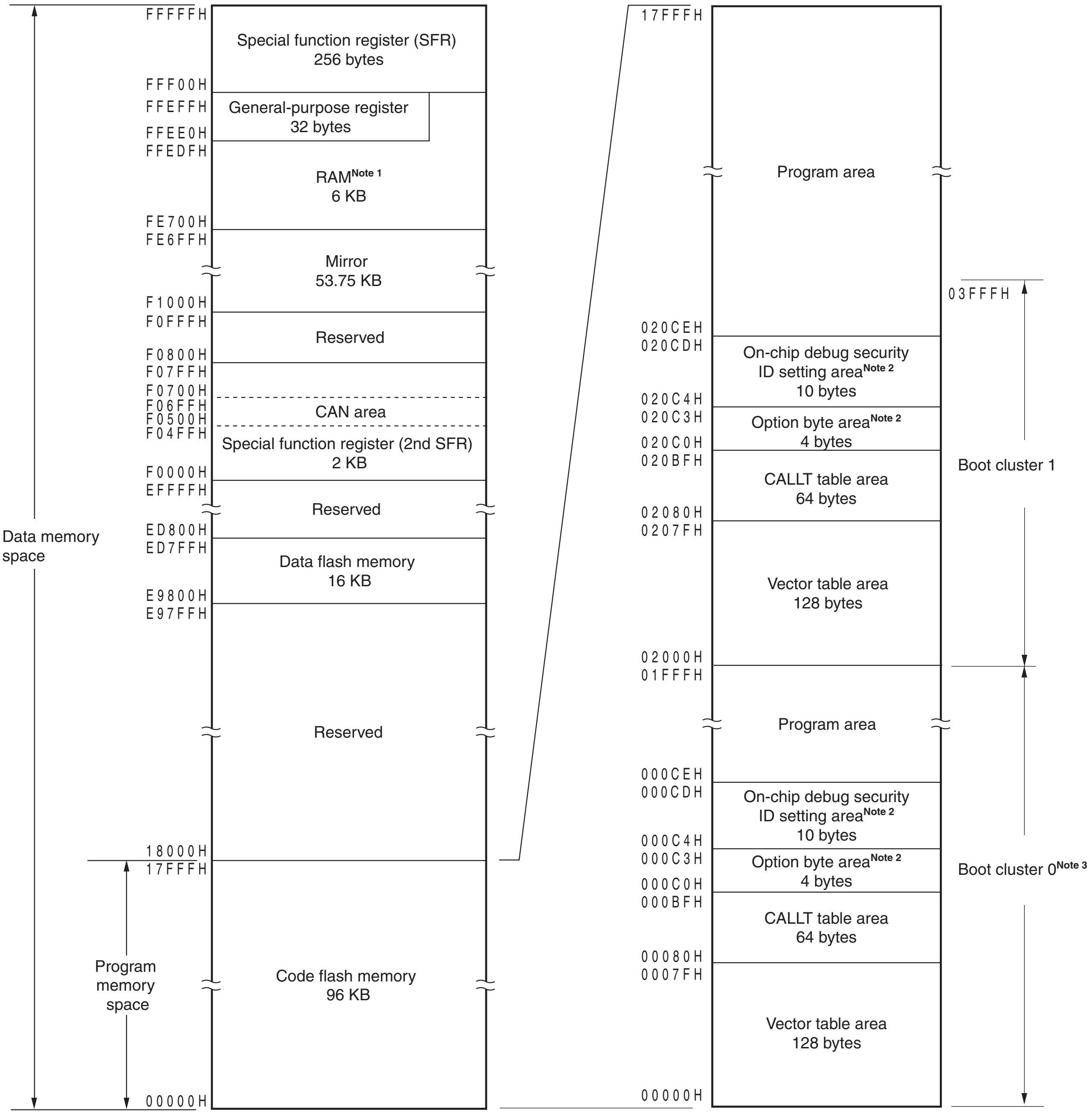
Notes 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

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 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **24.9 Security Setting**).

Figure 3-2. Memory Map (μPD78F1032, 78F1037, 78F1042, 78F1047)



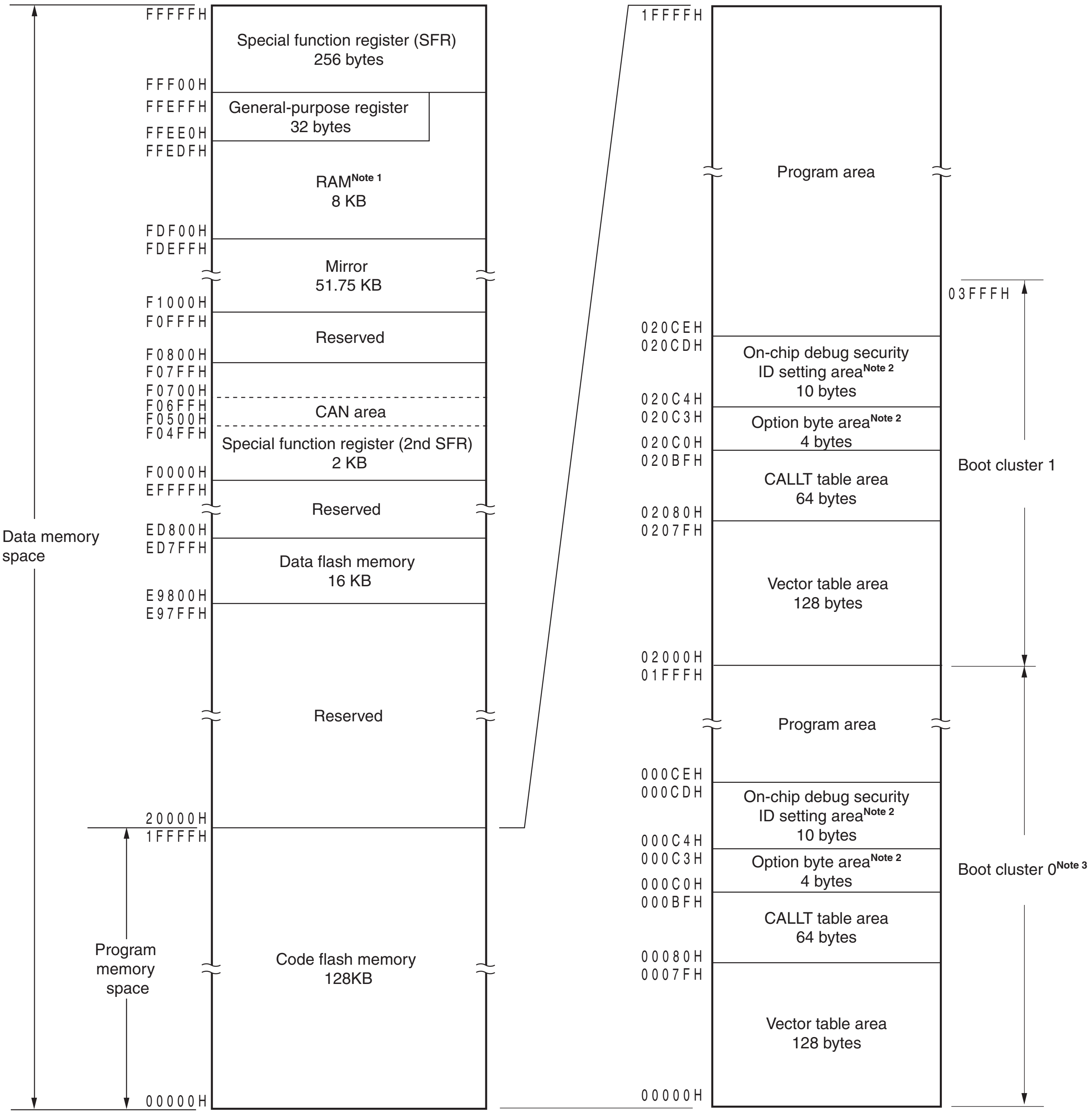
Notes 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

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 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **24.9 Security Setting**).

Figure 3-3. Memory Map (μPD78F1033, 78F1038, 78F1043, 78F1048)



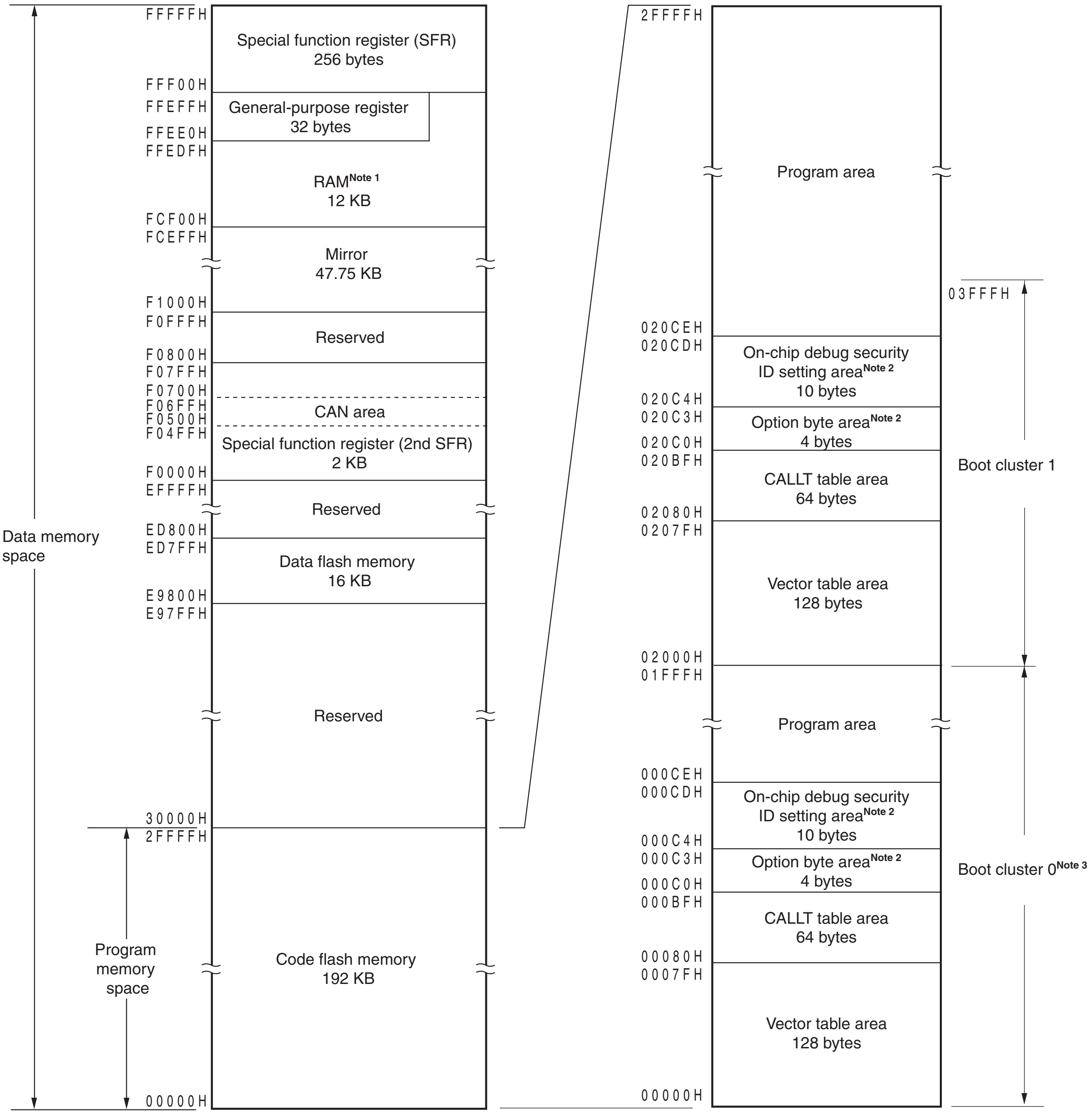
Notes 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

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 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **24.9 Security Setting**).

Figure 3-4. Memory Map (μPD78F1034, 78F1039, 78F1044, 78F1049)



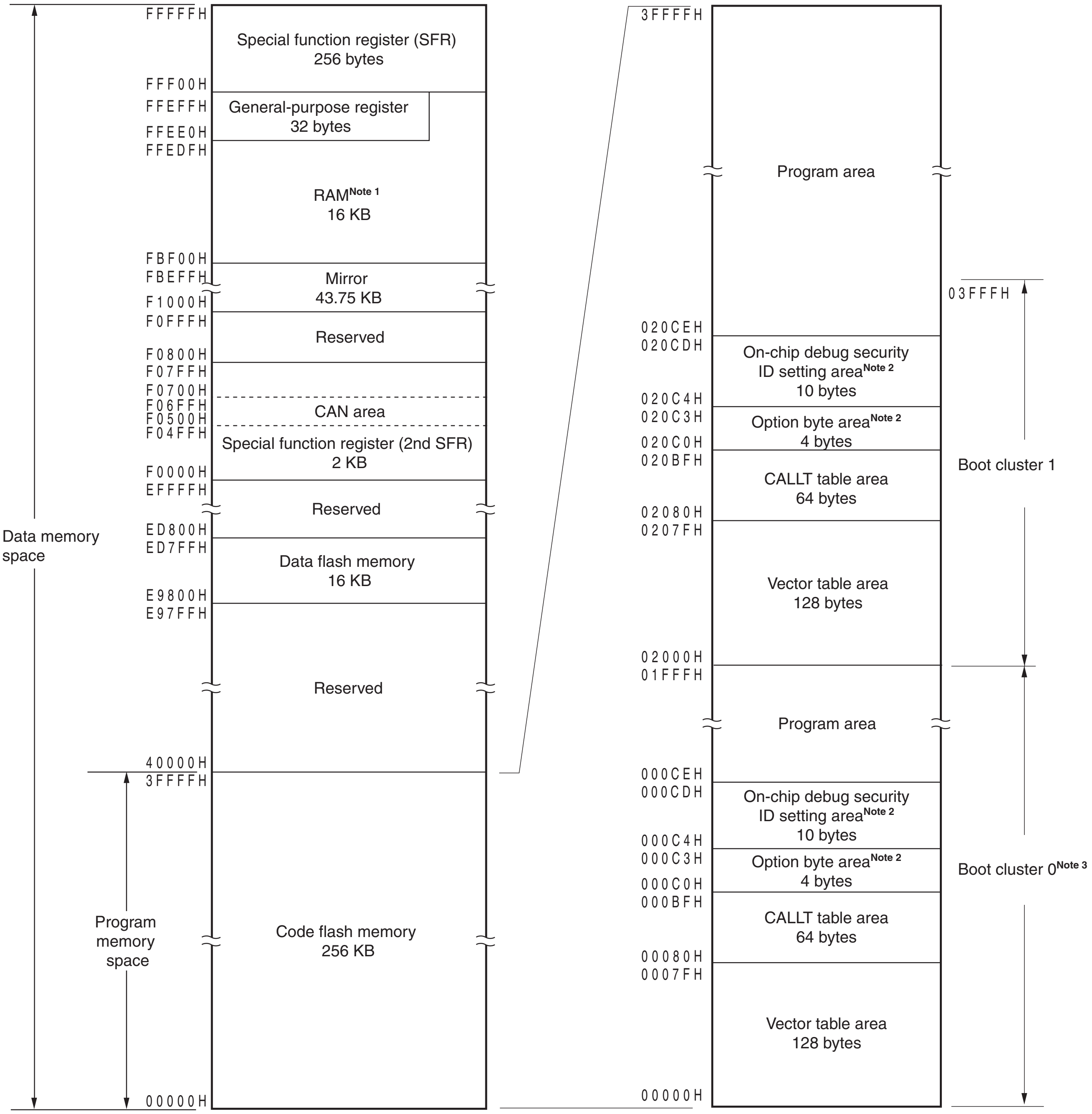
Notes 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

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 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **24.9 Security Setting**).

Figure 3-5. Memory Map (μPD78F1035, 78F1040, 78F1045, 78F1050)



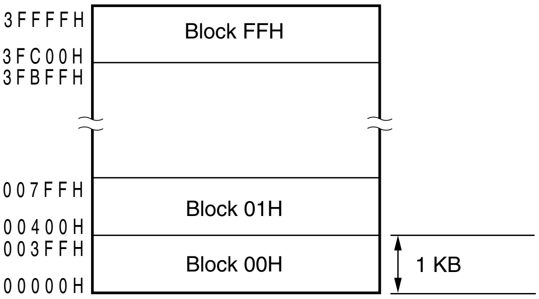
Notes 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

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 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **24.9 Security Setting**).

Remark The code 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 code flash memory are shown below.

Table 3-1. Correspondence Between Address Values and Block Numbers in Code Flash Memory (1/2)

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

Remark μPD78F1031, 78F1036, 78F1041, 78F1046: Block numbers 00H to 3FH
μPD78F1032, 78F1037, 78F1042, 78F1047: Block numbers 00H to 5FH
μPD78F1033, 78F1038, 78F1043, 78F1048: Block numbers 00H to 7FH

Table 3-1. Correspondence Between Address Values and Block Numbers in Code Flash Memory (2/2)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
20000H to 203FFH	80H	28000H to 283FFH	A0H	30000H to 303FFH	C0H	38000H to 383FFH	E0H
20400H to 207FFH	81H	28400H to 287FFH	A1H	30400H to 307FFH	C1H	38400H to 387FFH	E1H
20800H to 20BFFH	82H	28800H to 28BFFH	A2H	30800H to 30BFFH	C2H	38800H to 38BFFH	E2H
20C00H to 20FFFH	83H	28C00H to 29FFFH	A3H	30C00H to 30FFFH	C3H	38C00H to 39FFFH	E3H
21000H to 213FFH	84H	29000H to 293FFH	A4H	31000H to 313FFH	C4H	39000H to 393FFH	E4H
21400H to 217FFH	85H	29400H to 297FFH	A5H	31400H to 317FFH	C5H	39400H to 397FFH	E5H
21800H to 21BFFH	86H	29800H to 29BFFH	A6H	31800H to 31BFFH	C6H	39800H to 39BFFH	E6H
21C00H to 21FFFH	87H	29C00H to 29FFFH	A7H	31C00H to 31FFFH	C7H	39C00H to 39FFFH	E7H
22000H to 223FFH	88H	2A000H to 2A3FFH	A8H	32000H to 323FFH	C8H	3A000H to 3A3FFH	E8H
22400H to 227FFH	89H	2A400H to 2A7FFH	A9H	32400H to 327FFH	C9H	3A400H to 3A7FFH	E9H
22800H to 22BFFH	8AH	2A800H to 2ABFFH	AAH	32800H to 32BFFH	CAH	3A800H to 3ABFFH	EAH
22C00H to 22FFFH	8BH	2AC00H to 2AFFFH	ABH	32C00H to 32FFFH	CBH	3AC00H to 3AFFFH	EBH
23000H to 233FFH	8CH	2B000H to 2B3FFH	ACH	33000H to 333FFH	CCH	3B000H to 3B3FFH	ECH
23400H to 237FFH	8DH	2B400H to 2B7FFH	ADH	33400H to 337FFH	CDH	3B400H to 3B7FFH	EDH
23800H to 23BFFH	8EH	2B800H to 2BBFFH	AEH	33800H to 33BFFH	CEH	3B800H to 3BBFFH	EEH
23C00H to 23FFFH	8FH	2BC00H to 2BFFFH	AFH	33C00H to 33FFFH	CFH	3BC00H to 3BFFFH	EFH
24000H to 243FFH	90H	2C000H to 2C3FFH	B0H	34000H to 343FFH	D0H	3C000H to 3C3FFH	F0H
24400H to 247FFH	91H	2C400H to 2C7FFH	B1H	34400H to 347FFH	D1H	3C400H to 3C7FFH	F1H
24800H to 24BFFH	92H	2C800H to “CBFFH	B2H	34800H to 34BFFH	D2H	3C800H to 3CBFFH	F2H
24C00H to 24FFFH	93H	2CC00H to 2CFFFH	B3H	34C00H to 34FFFH	D3H	3CC00H to 3CFFFH	F3H
25000H to 253FFH	94H	2D000H to 2D3FFH	B4H	35000H to 353FFH	D4H	3D000H to 3D3FFH	F4H
25400H to 257FFH	95H	2D400H to 2D7FFH	B5H	35400H to 357FFH	D5H	3D400H to 3D7FFH	F5H
25800H to 25BFFH	96H	2D800H to 2DBFFH	B6H	35800H to 35BFFH	D6H	3D800H to 3DBFFH	F6H
25C00H to 25FFFH	97H	2DC00H to 2DFFFH	B7H	35C00H to 35FFFH	D7H	3DC00H to 3DFFFH	F7H
26000H to 263FFH	98H	2E000H to 2E3FFH	B8H	36000H to 363FFH	D8H	3E000H to 3E3FFH	F8H
26400H to 267FFH	99H	2E400H to 2E7FFH	B9H	36400H to 367FFH	D9H	3E400H to 3E7FFH	F9H
26800H to 26BFFH	9AH	2E800H to 2EBFFH	BAH	36800H to 36BFFH	DAH	3E800H to 3EBFFH	FAH
26C00H to 26FFFH	9BH	2EC00H to 2EFFFH	BBH	36C00H to 36FFFH	DBH	3EC00H to 3EFFFH	FBH
27000H to 273FFH	9CH	2F000H to 2F3FFH	BCH	37000H to 373FFH	DCH	3F000H to 3F3FFH	FCH
27400H to 277FFH	9DH	2F400H to 2F7FFH	BDH	37400H to 377FFH	DDH	3F400H to 3F7FFH	FDH
27800H to 27BFFH	9EH	2F800H to 2FBFFH	BEH	37800H to 37BFFH	DEH	3F800H to 3FBFFH	FEH
27C00H to 27FFFH	9FH	2FC00H to 2FFFFH	BFH	37C00H to 37FFFH	DFH	3FC00H to 3FFFFH	FFH

Remark μPD78F1034, 78F1039, 78F1044, 78F1049: Block numbers 00H to BFH
μPD78F1035, 78F1040, 78F1045, 78F1050: Block numbers 00H to FFH

3.1.1 Internal program memory space

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

78K0R/Hx3 products incorporate internal ROM (code flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F1031, 78F1036, 78F1041, 78F1046	Code flash memory	65536 × 8 bits (00000H to 0FFFFH)
μPD78F1032, 78F1037, 78F1042, 78F1047		98304 × 8 bits (00000H to 17FFFH)
μPD78F1033, 78F1038, 78F1043, 78F1048		131072 × 8 bits (00000H to 1FFFFH)
μPD78F1034, 78F1039, 78F1044, 78F1049		196608 × 8 bits (00000H to 2FFFFH)
μPD78F1035, 78F1040, 78F1045, 78F1050		262144 × 8 bits (00000H to 3FFFFH)

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, IAW, CLKM	00040H	INTCSI11, INTIIC11
		00042H	INTTM04
00004H	INTWDTI	00044H	INTTM05
00006H	INTLVI	00046H	INTTM06
00008H	INTP0	00048H	INTTM07
0000AH	INTP1	0004AH	INTP6/INTKR
0000CH	INTP2	0004CH	INTP7
0000EH	INTP3	0004EH	INTC0ERR
00010H	INTP4	00050H	INTC0WUP
00012H	INTP5	00052H	INTC0REC
00014H	INTCLM	00054H	INTC0TRX
00016H	INTCSI00	00056H	INTTM10
00018H	INTCSI01	00058H	INTTM11
0001AH	INTDMA0	0005AH	INTTM12
0001CH	INTDMA1	0005CH	INTTM13
0001EH	INTWUTM	0005EH	INTMD
00020H	INTFL	00060H	INTST2/INTIIC20
00022H	INTLT0	00062H	INTSR2
00024H	INTLR0	00064H	INTPR2
00026H	INTLS0	00066H	INTTM14
00028H	INTPLR0	00068H	INTTM15
0002AH	INTP8	0006AH	INTTM16
0002CH	INTTM00	0006CH	INTTM17
0002EH	INTTM01	0006EH	INTTM20
00030H	INTTM02	00070H	INTTM21
00032H	INTTM03	00072H	INTTM22
00034H	INTAD	00074H	INTTM23
00036H	INTLT1	00076H	INTTM25
00038H	INTLR1	00078H	INTTM27
0003AH	INTLS1	0007AH	INTDMA2
0003CH	INTPLR1	0007CH	INTDMA3
0003EH	INTCSI10	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 23 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 26 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The 78K0R/Hx3 mirrors the code 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)).

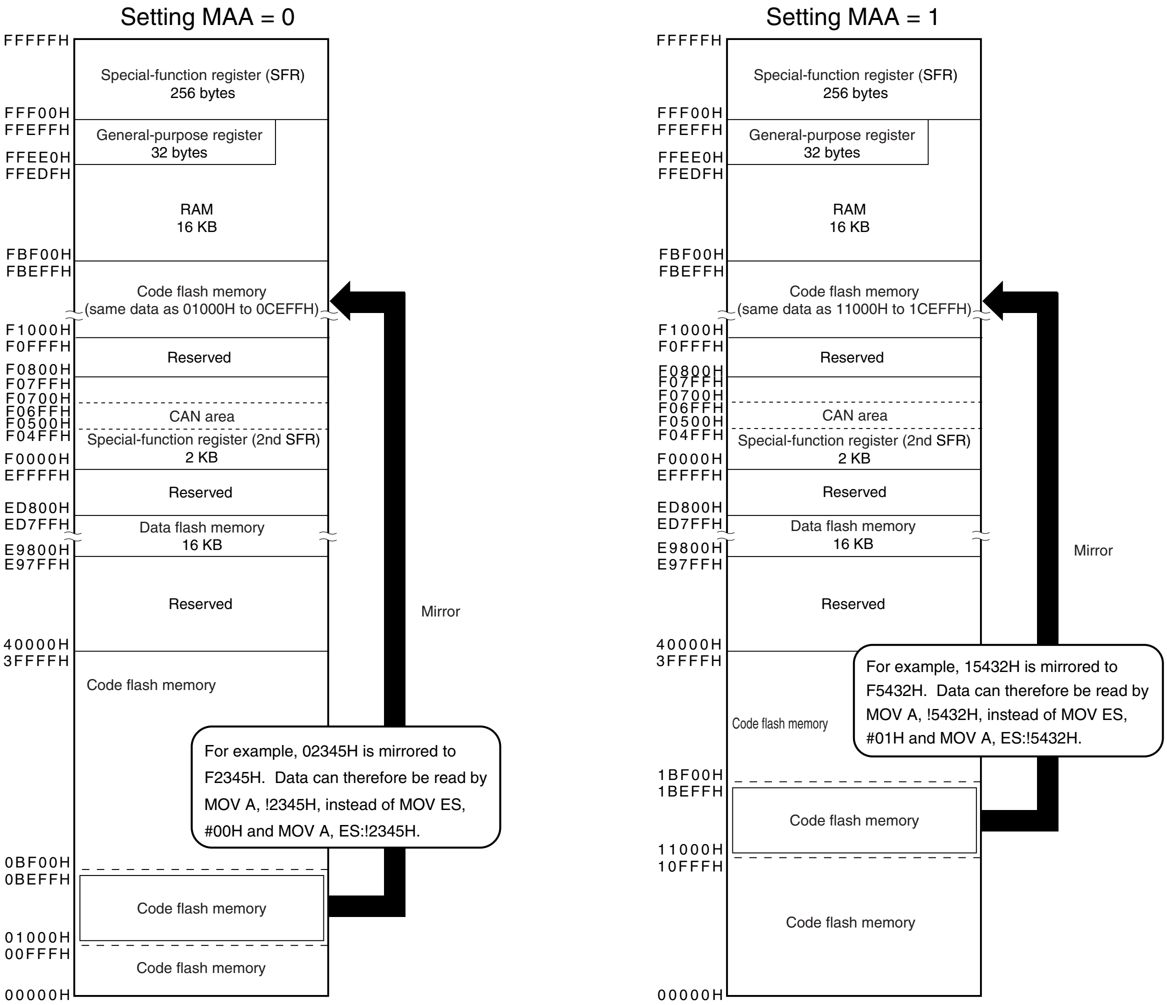
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the data flash can be read with the shorter code. However, the data flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

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

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

The following show examples.

Example μPD78F1035, 78F1040, 78F1045, 78F1050 (Code flash memory: 256 KB, RAM: 16 KB)



For example, 02345H is mirrored to F2345H. Data can therefore be read by MOV A, !2345H, instead of MOV ES, #00H and MOV A, ES:!2345H.

For example, 15432H is mirrored to F5432H. Data can therefore be read by MOV A, !5432H, instead of MOV ES, #01H and MOV A, ES:!5432H.

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 code 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 3-6. 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 code 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. Be sure to clear bit 0 (MAA) to 0 for products with the code flash memory of no more than 64 KB.

3.1.3 Internal data memory space

78K0R/Hx3 products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
μPD78F1031, 78F1036, 78F1041, 78F1046	4096 × 8 bits (FEF00H to FFEFFH)
μPD78F1032, 78F1037, 78F1042, 78F1047	6144 × 8 bits (FE700H to FFEFFH)
μPD78F1033, 78F1038, 78F1043, 78F1048	8192 × 8 bits (FDF00H to FFEFFH)
μPD78F1034, 78F1039, 78F1044, 78F1049	12288 × 8 bits (FCF00H to FFEFFH)
μPD78F1035, 78F1040, 78F1045, 78F1050	16384 × 8 bits (FBF00H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are written and executed. Four general-purpose register banks consisting of eight 8-bit registers per bank are assigned to the 32-byte area of FFEE0H to FFEFFH of the internal RAM area. However, instructions cannot be executed by using the general-purpose registers.

The internal RAM is used as a stack memory.

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

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FFF00H to FFFFFH (see **Table 3-5** in **3.2.4 Special function registers (SFRs)**).

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

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

On-chip peripheral hardware special function registers (2nd SFRs) are allocated in the area F0000H to F07FFH (see **Table 3-6** in **3.2.5 Extended Special function registers (2nd SFRs: 2nd Special Function Registers)**).

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

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

3.1.6 Data memory addressing

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

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the 78K0R/Hx3, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Figures 3-7 to 3-11 show correspondence between data memory and addressing.

Figure 3-7. Correspondence Between Data Memory and Addressing (μPD78F1031, 78F1036, 78F1041, 78F1046)

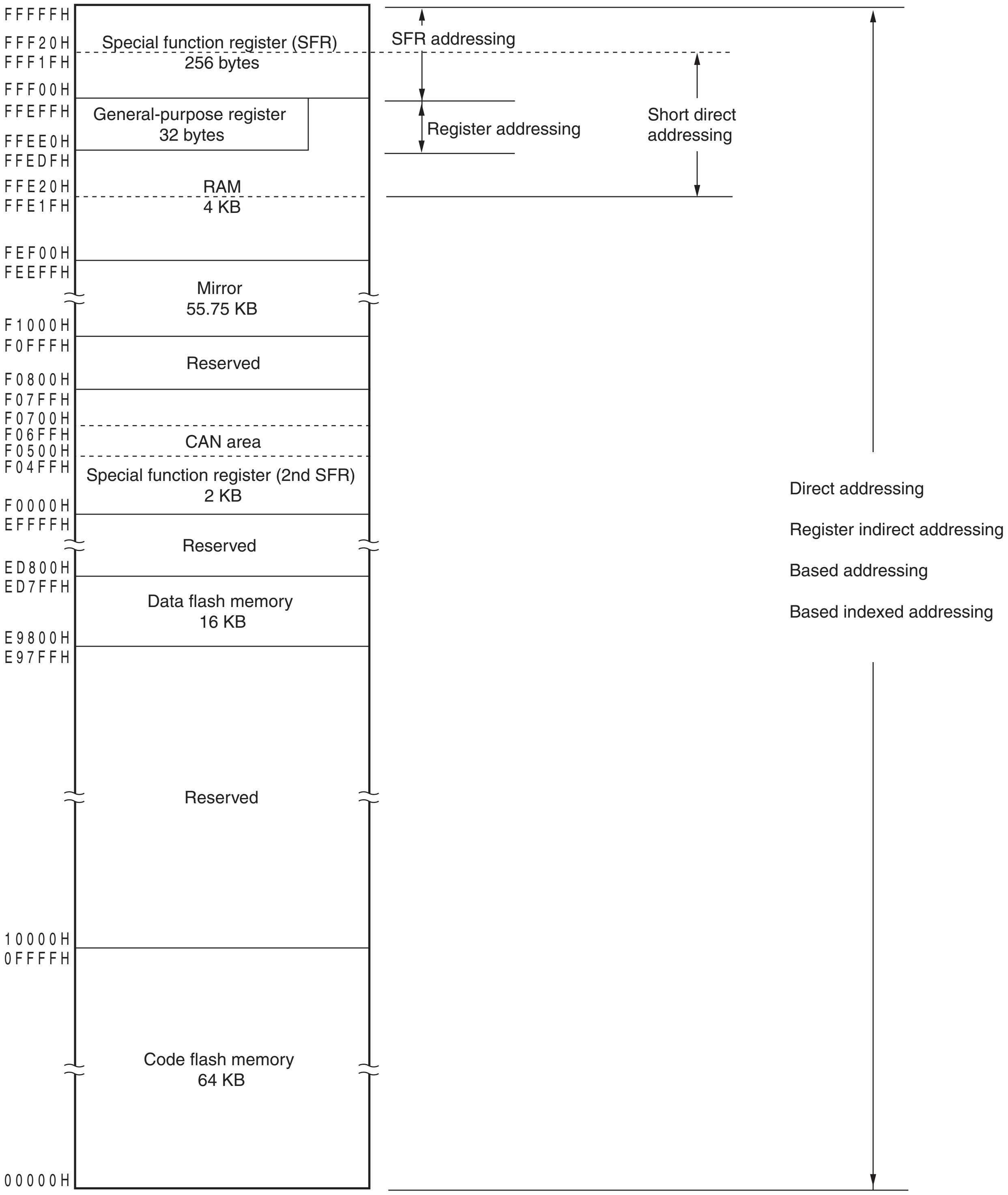


Figure 3-8. Correspondence Between Data Memory and Addressing (μPD78F1032, 78F1037, 78F1042, 78F1047)

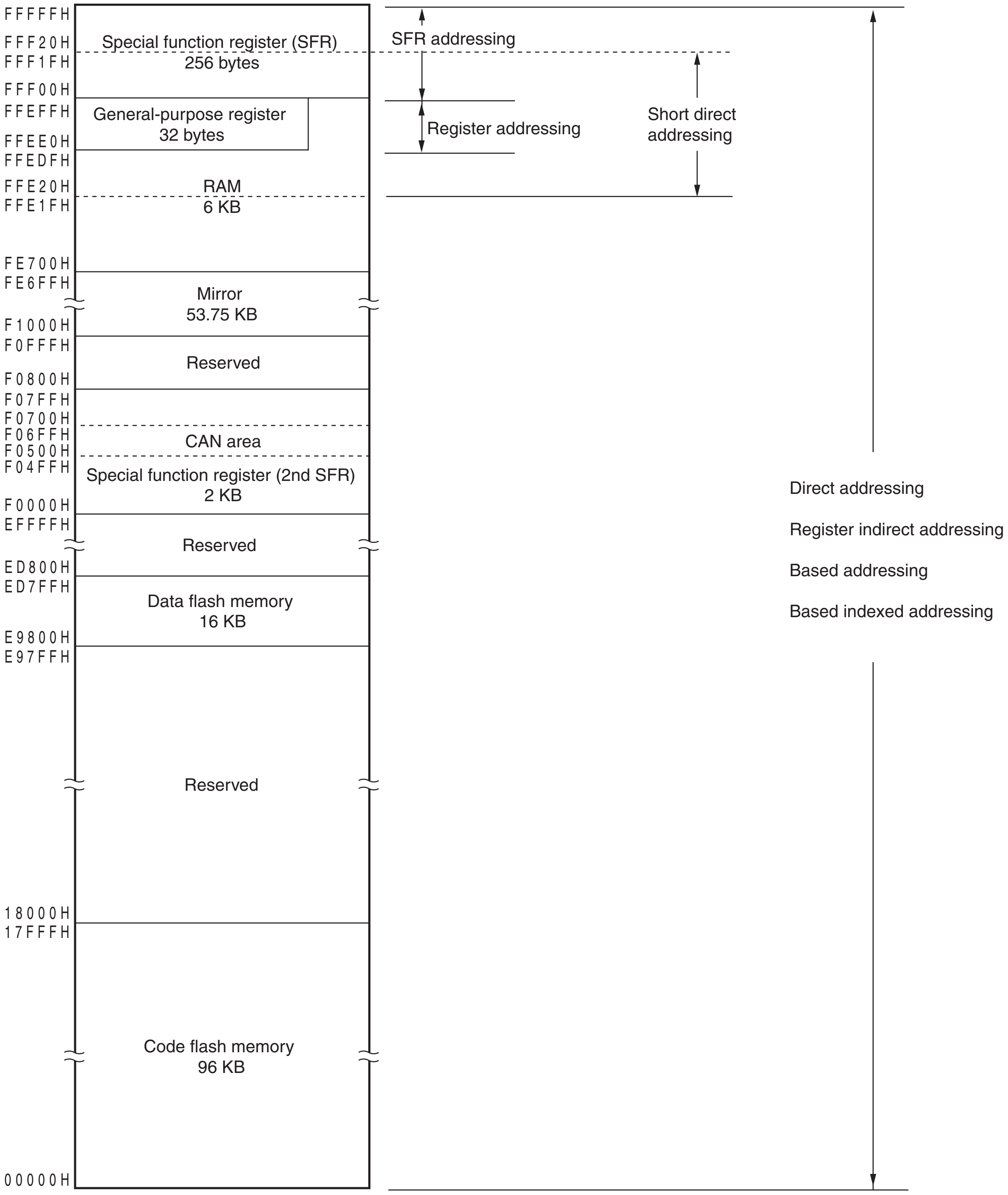


Figure 3-9. Correspondence Between Data Memory and Addressing (μPD78F1033, 78F1038, 78F1043, 78F1048)

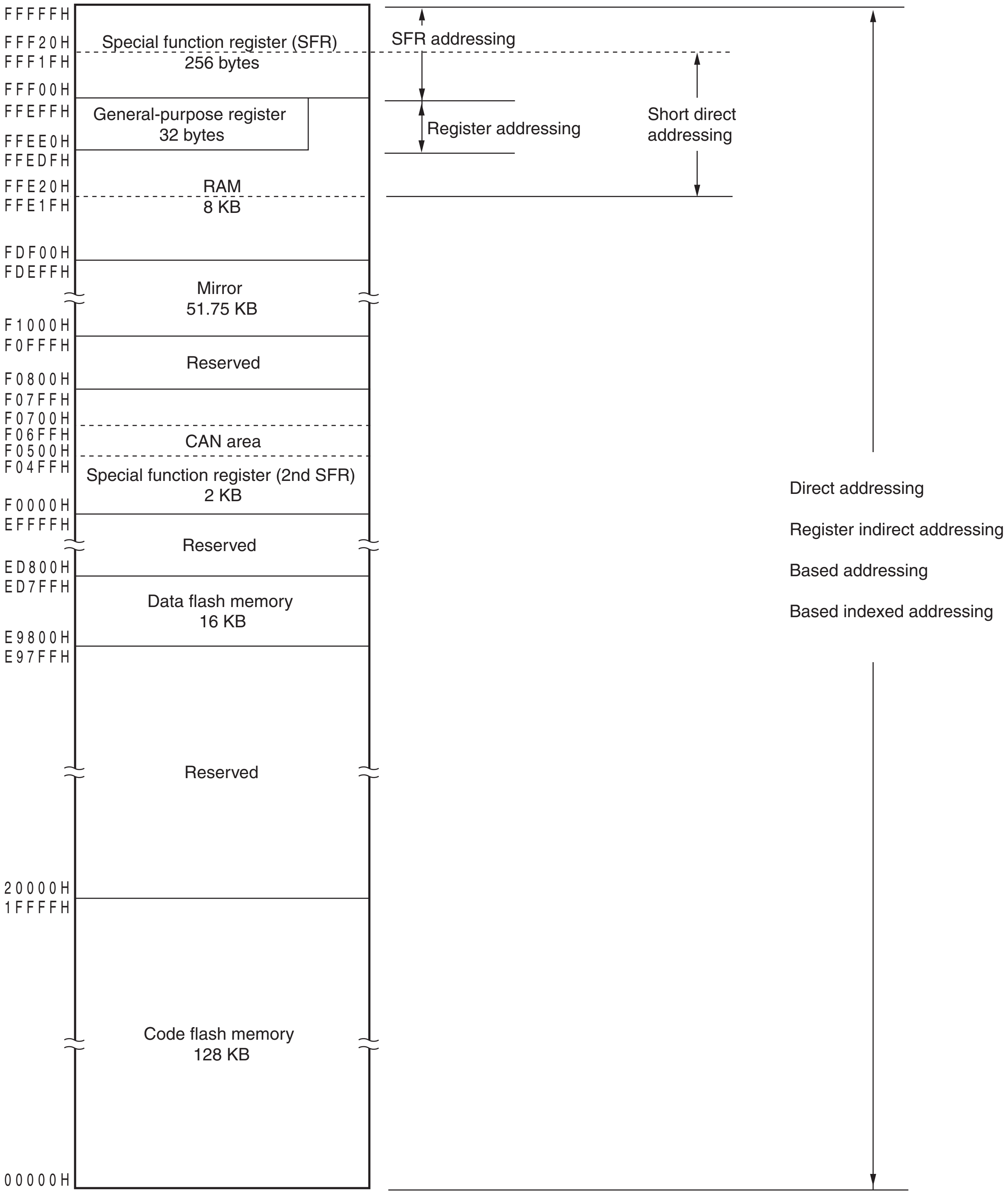


Figure 3-10. Correspondence Between Data Memory and Addressing (μPD78F1034, 78F1039, 78F1044, 78F1049)

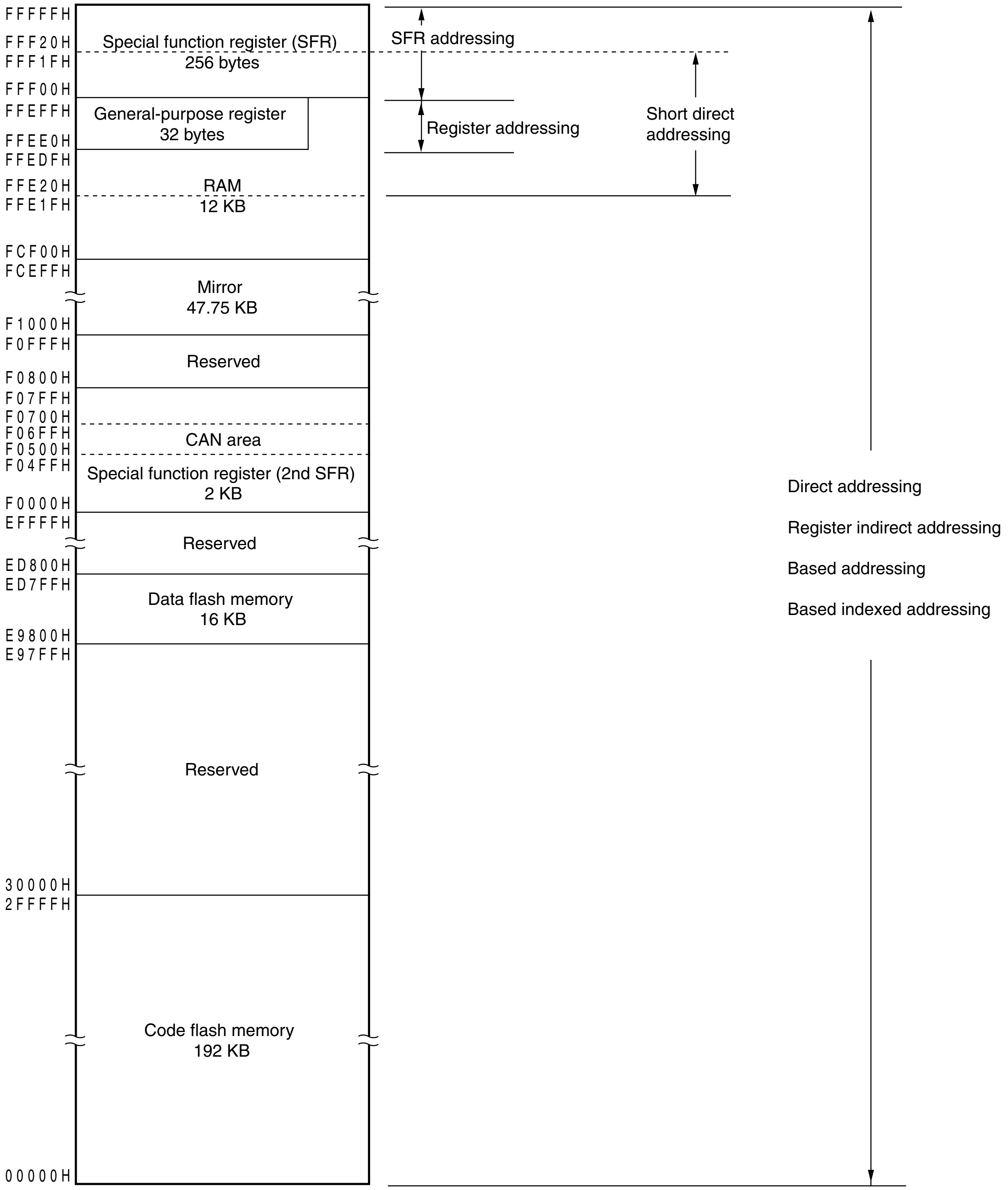
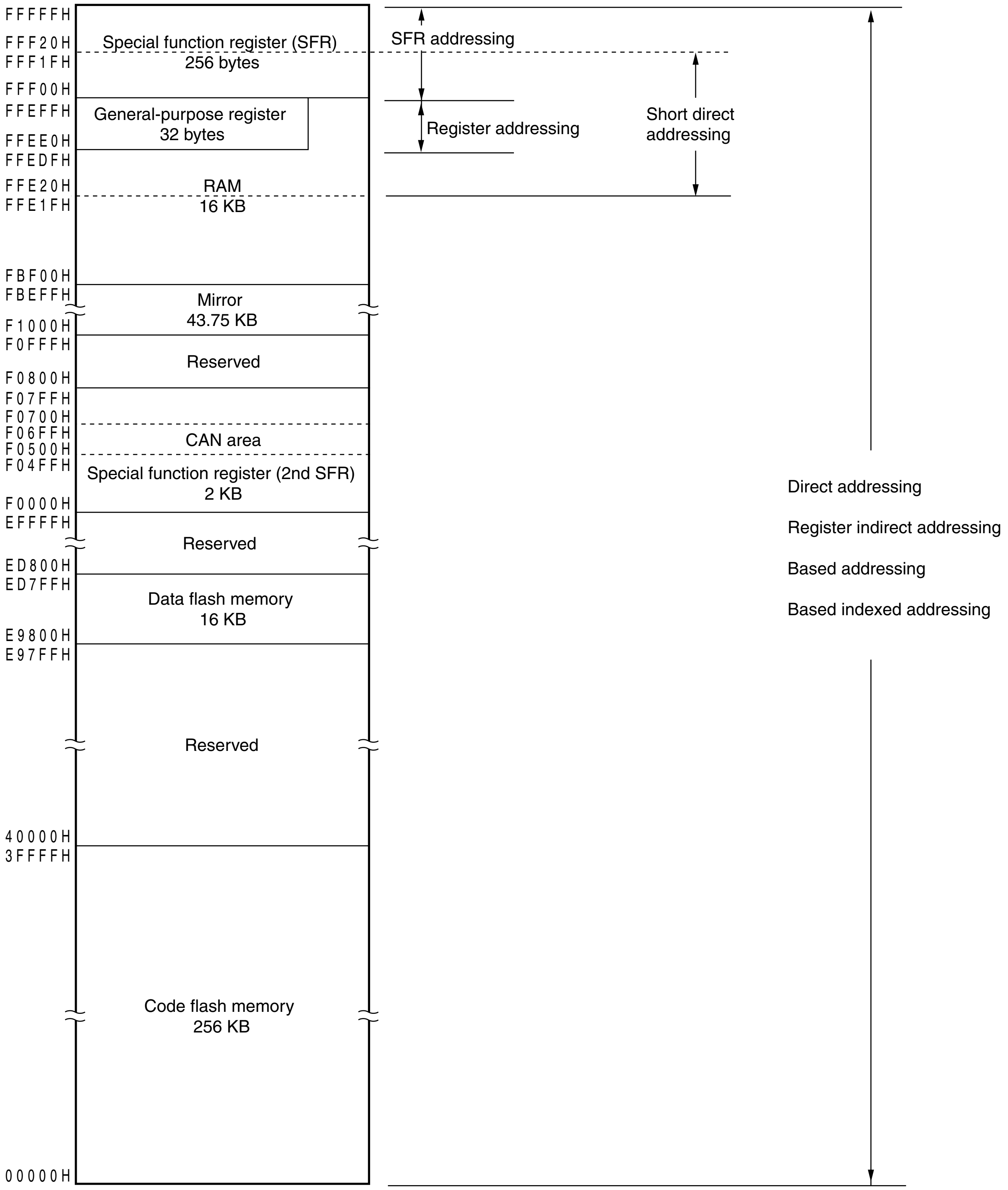


Figure 3-11. Correspondence Between Data Memory and Addressing (μPD78F1035, 78F1040, 78F1045, 78F1050)



3.2 Processor Registers

The 78K0R/Hx3 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 3-12. Format of Program Counter

	19	0
PC		

(2) Program status word (PSW)

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

Program status word contents are stored in the stack area upon vectored interrupt request is acknowledged or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions. Reset signal generation sets the PSW register to 06H.

Figure 3-13. 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, PRn3L, PRn3H) (see **16.3 (3)**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-14. 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.

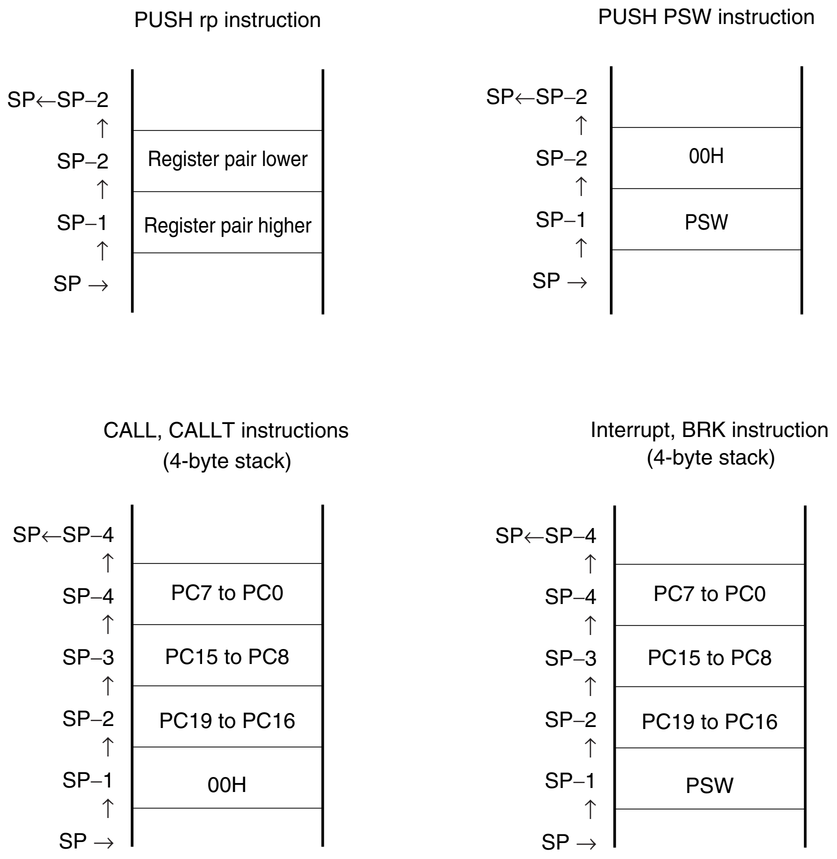
Each stack operation saves data as shown in Figure 3-15.

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

2. The values of the stack pointer must be set to even numbers. If odd numbers are specified, the least significant bit is automatically cleared to 0.

3. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as a stack area.

Figure 3-15. Data to Be Saved to Stack Memory



PUSH rp instruction

PUSH PSW instruction

CALL, CALLT instructions (4-byte stack)

Interrupt, BRK instruction (4-byte stack)

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).

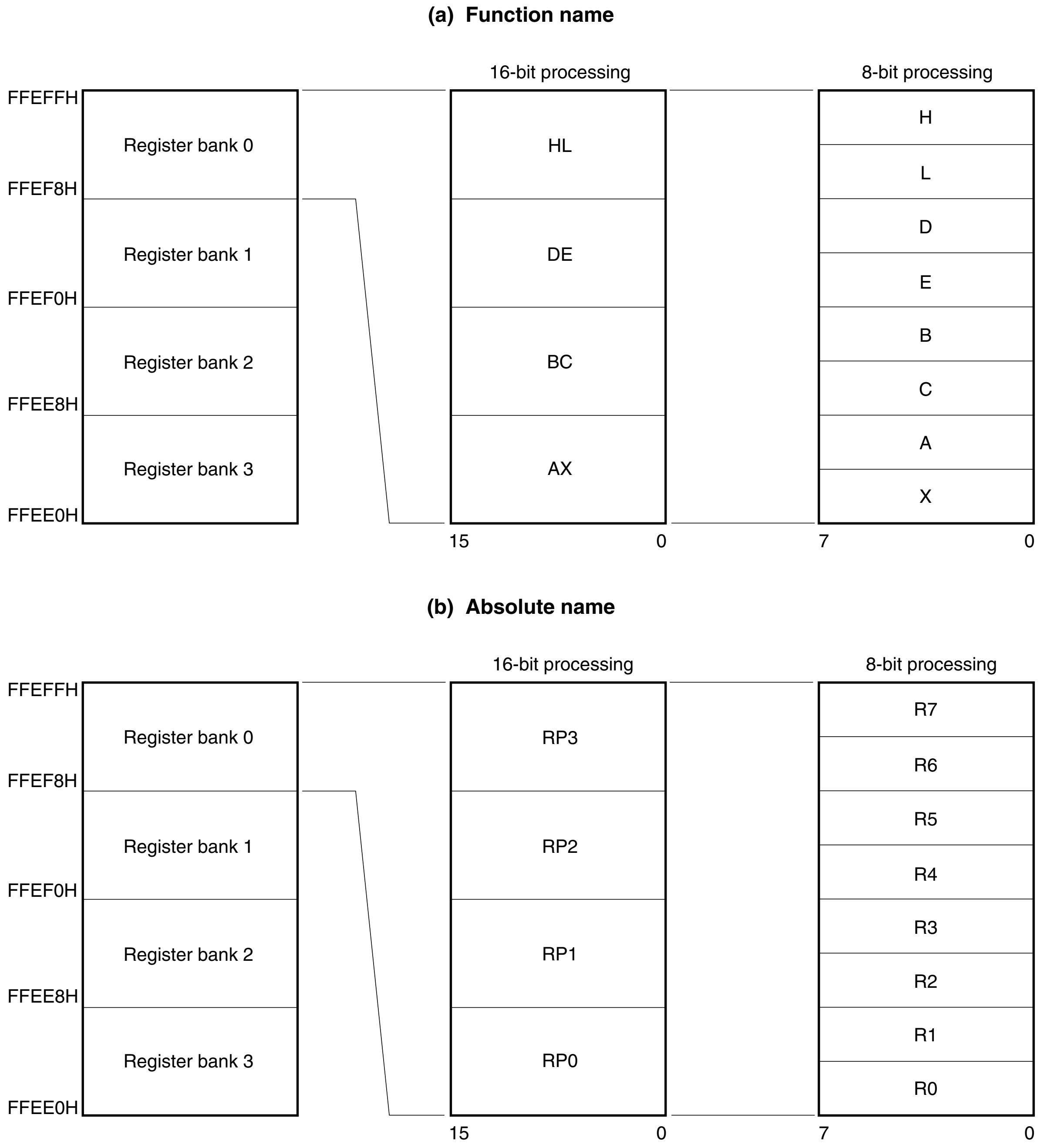
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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 3-16. 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 3-17. 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 extended SFRs are not assigned.

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

Table 3-5. SFR List (1/6)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
					1-bit	8-bit	16-bit		48	64	80	100
FFF00H	Port register 0	P0		R/W	√	√	–	00H	√	√	√	√
FFF01H	Port register 1	P1		R/W	√	√	–	00H	√	√	√	√
FFF03H	Port register 3	P3		R/W	√	√	–	00H	√	√	√	√
FFF04H	Port register 4	P4		R/W	√	√	–	00H	√	√	√	√
FFF05H	Port register 5	P5		R/W	√	√	–	00H	–	√	√	√
FFF06H	Port register 6	P6		R/W	√	√	–	00H	√	√	√	√
FFF07H	Port register 7	P7		R/W	√	√	–	00H	√	√	√	√
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	√	√	–	00H	√	√	√	√
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	SDR00L	SDR00	R/W	–	√	√	0000H	√	√	√	√
FFF11H		–			–	–						
FFF12H	Serial data register 01	SDR01L	SDR01	R/W	–	√	√	0000H	–	√	√	√
FFF13H		–			–	–						
FFF14H	Serial data register 10	SDR10L	SDR10	R/W	–	√	√	0000H	√	√	√	√
FFF15H		–			–	–						
FFF16H	Serial data register 11	SDR11L	SDR11	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	√	√	√	√

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
				1-bit	8-bit	16-bit		48	64	80	100
FFF20H	Port mode register 0	PM0	R/W	√	√	–	FFH	√	√	√	√
FFF21H	Port mode register 1	PM1	R/W	√	√	–	FFH	√	√	√	√
FFF23H	Port mode register 3	PM3	R/W	√	√	–	FFH	√	√	√	√
FFF24H	Port mode register 4	PM4	R/W	√	√	–	FFH	√	√	√	√
FFF25H	Port mode register 5	PM5	R/W	√	√	–	FFH	–	√	√	√
FFF26H	Port mode register 6	PM6	R/W	√	√	–	FFH	√	√	√	√
FFF27H	Port mode register 7	PM7	R/W	√	√	–	FFH	√	√	√	√
FFF28H	Port mode register 8	PM8	R/W	√	√	–	FFH	√	√	√	√
FFF29H	Port mode register 9	PM9	R/W	√	√	–	FFH	√	√	√	√
FFF2AH	Port mode register 10	PM10	R/W	√	√	–	FFH	–	–	–	√
FFF2CH	Port mode register 12	PM12	R/W	√	√	–	FFH	√	√	√	√
FFF2EH	Port mode register 14	PM14	R/W	√	√	–	FEH	√	√	√	√
FFF2FH	Port mode register 15	PM15	R/W	√	√	–	FFH	–	–	–	√
FFF30H	A/D converter mode register 0	ADM0	R/W	√	√	–	00H	√	√	√	√
FFF31H	Analog input channel specification register	ADS	R/W	√	√	–	00H	√	√	√	√
FFF33H	A/D conversion time setting register	ADSMP	R/W	√	√	–	00H	√	√	√	√
FFF36H	External interrupt input pin selection register 0	IPSEL0	R/W	√	√	–	00H	√	√	√	√
FFF37H	Key return mode register	KRM	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	Serial communication pin select register	STSEL	R/W	√	√	–	00H	√	√	√	√
FFF3EH	Timer input select register 0	TIS0	R/W	√	√	–	00H	√	√	√	√
FFF3FH	Timer input select register 1	TIS1	R/W	√	√	–	00H	√	√	√	√
FFF42H	A/D converter mode register 1	ADM1	R/W	√	√	–	00H	√	√	√	√

Table 3-5. SFR List (3/6)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
					1-bit	8-bit	16-bit		48	64	80	100
FFF44H	Serial data register 20	SDR20L	SDR20	R/W	–	√	√	0000H	–	√	√	√
FFF45H		–			–	–						
FFF46H	Serial data register 21	SDR21L	SDR21	R/W	–	√	√	0000H	–	√	√	√
FFF47H		–			–	–						
FFF48H	LIN-UART0 8-bit transmit data register	UF0TXB	UF0TX	R/W	–	√	√	00H	√	√	√	√
FFF49H	LIN-UART0 transmit data register	–			–	–		0000H	√	√	√	√
FFF4AH	LIN-UART0 8-bit receive data register	UF0RXB	UF0RX	R	–	√	√	00H	√	√	√	√
FFF4BH	LIN-UART0 receive data register	–			–	–		0000H	√	√	√	√
FFF4CH	LIN-UART1 8-bit transmit data register	UF1TXB	UF1TX	R/W	–	√	√	00H	√	√	√	√
FFF4DH	LIN-UART1 transmit data register	–			–	–		0000H	√	√	√	√
FFF4EH	LIN-UART1 8-bit receive data register	UF1RXB	UF1RX	R	–	√	√	00H	√	√	√	√
FFF4FH	LIN-UART1 receive data register	–			–	–		0000H	√	√	√	√
FFF60H	Timer output select register 0	TOS0		R/W	√	√	–	00H	√	√	√	√
FFF61H	Timer output select register 1	TOS1		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												
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												

Table 3-5. SFR List (4/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
				1-bit	8-bit	16-bit		48	64	80	100
FFF78H	Timer data register 14	TDR14	R/W	–	–	√	0000H	√	√	√	√
FFF79H											
FFF7AH	Timer data register 15	TDR15	R/W	–	–	√	0000H	√	√	√	√
FFF7BH											
FFF7CH	Timer data register 16	TDR16	R/W	–	–	√	0000H	√	√	√	√
FFF7DH											
FFF7EH	Timer data register 17	TDR17	R/W	–	–	√	0000H	√	√	√	√
FFF7FH											
FFF90H	Timer data register 20	TDR20	R/W	–	–	√	0000H	–	√	√	√
FFF91H											
FFF92H	Timer data register 21	TDR21	R/W	–	–	√	0000H	–	√	√	√
FFF93H											
FFF94H	Timer data register 22	TDR22	R/W	–	–	√	0000H	–	√	√	√
FFF95H											
FFF96H	Timer data register 23	TDR23	R/W	–	–	√	0000H	–	√	√	√
FFF97H											
FFF98H	Timer data register 24	TDR24	R/W	–	–	√	0000H	–	–	–	√
FFF99H											
FFF9AH	Timer data register 25	TDR25	R/W	–	–	√	0000H	–	–	–	√
FFF9BH											
FFF9CH	Timer data register 26	TDR26	R/W	–	–	√	0000H	–	–	–	√
FFF9DH											
FFF9EH	Timer data register 27	TDR27	R/W	–	–	√	0000H	–	–	–	√
FFF9FH											
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	CKS	R/W	√	√	–	00H[Note 1]	√	√	√	√
FFFA8H	Reset control flag register	RESF	R	–	√	–	00H[Note 2]	√	√	√	√
FFFA9H	Low-voltage detection register	LVIM	R/W	√	√	–	00H[Note 3]	√	√	√	√

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

2. The reset value of RESF varies depending on the reset source.

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

Table 3-5. SFR List (5/6)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
					1-bit	8-bit	16-bit		48	64	80	100
FFFAAH	Low-voltage detection level select register	LVIS		R/W	√	√	–	09H[Note 1]	√	√	√	√
FFFABH	Watchdog timer enable register	WDTE		R/W	–	√	–	1A/ 9A[Note 2]	√	√	√	√
FFFACH	WUTM control register	WUTMCTL		R/W	√	√	–	00H	√	√	√	√
FFFAEH	WUTM compare register	WUTMCMP		R/W	–	–	√	0000H	√	√	√	√
FFFAFH												
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	√	√	√	√
FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	√	√	√	00H	√	√	√	√
FFFD1H	Interrupt request flag register 2H	IF2H		R/W	√	√		00H	√	√	√	√
FFFD2H	Interrupt request flag register 3L	IF3L	IF3	R/W	√	√	√	00H	√	√	√	√
FFFD3H	Interrupt request flag register 3H	IF3H		R/W	√	√		00H	√	√	√	√
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH	√	√	√	√
FFFD5H	Interrupt mask flag register 2H	MK2H		R/W	√	√		FFH	√	√	√	√
FFFD6H	Interrupt mask flag register 3L	MK3L	MK3	R/W	√	√	√	FFH	√	√	√	√
FFFD7H	Interrupt mask flag register 3H	MK3H		R/W	√	√		FFH	√	√	√	√
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH	√	√	√	√
FFFD9H	Priority specification flag register 02H	PR02H		R/W	√	√		FFH	√	√	√	√
FFFDAH	Priority specification flag register 03L	PR03L	PR03	R/W	√	√	√	FFH	√	√	√	√
FFFDBH	Priority specification flag register 03H	PR03H		R/W	√	√		FFH	√	√	√	√
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH	√	√	√	√
FFFDDH	Priority specification flag register 12H	PR12H		R/W	√	√		FFH	√	√	√	√

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

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

Table 3-5. SFR List (6/6)

Address	Special Function Register (SFR) Name	Symbol		R/W		Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
						1-bit	8-bit	16-bit		48	64	80	100
FFFDEH	Priority specification flag register 13L	PR13L	PR13	R/W		√	√	√	FFH	√	√	√	√
FFFDFH	Priority specification flag register 13H	PR13H		R/W		√	√		FFH	√	√	√	√
FFFE0H	Interrupt request flag register 0L	IF0L	IF0	R/W		√	√	√	00H	√	√	√	√
FFFE1H	Interrupt request flag register 0H	IF0H		R/W		√	√		00H	√	√	√	√
FFFE2H	Interrupt request flag register 1L	IF1L	IF1	R/W		√	√	√	00H	√	√	√	√
FFFE3H	Interrupt request flag register 1H	IF1H		R/W		√	√		00H	√	√	√	√
FFFE4H	Interrupt mask flag register 0L	MK0L	MK0	R/W		√	√	√	FFH	√	√	√	√
FFFE5H	Interrupt mask flag register 0H	MK0H		R/W		√	√		FFH	√	√	√	√
FFFE6H	Interrupt mask flag register 1L	MK1L	MK1	R/W		√	√	√	FFH	√	√	√	√
FFFE7H	Interrupt mask flag register 1H	MK1H		R/W		√	√		FFH	√	√	√	√
FFFE8H	Priority specification flag register 00L	PR00L	PR00	R/W		√	√	√	FFH	√	√	√	√
FFFE9H	Priority specification flag register 00H	PR00H		R/W		√	√		FFH	√	√	√	√
FFFEAH	Priority specification flag register 01L	PR01L	PR01	R/W		√	√	√	FFH	√	√	√	√
FFFEBH	Priority specification flag register 01H	PR01H		R/W		√	√		FFH	√	√	√	√
FFFECH	Priority specification flag register 10L	PR10L	PR10	R/W		√	√	√	FFH	√	√	√	√
FFFEDH	Priority specification flag register 10H	PR10H		R/W		√	√		FFH	√	√	√	√
FFFEEH	Priority specification flag register 11L	PR11L	PR11	R/W		√	√	√	FFH	√	√	√	√
FFFEFH	Priority specification flag register 11H	PR11H		R/W		√	√		FFH	√	√	√	√
FFFF0H	Multiplication input data register A (L)	MDAL/MULA		R/W	R/W	–	–	√	0000H	√	√	√	√
FFFF1H													
FFFF2H	Multiplication input data register A (H)	MDAH/MULB		R/W	R/W	–	–	√	0000H	√	√	√	√
FFFF3H													
FFFF4H	Multiplication input data register B (H)	MDBH/MULOH		R/W	R	–	–	√	0000H	√	√	√	√
FFFF5H													
FFFF6H	Multiplication input data register B (L)	MDBL/MULOL		R/W	R	–	–	√	0000H	√	√	√	√
FFFF7H													
FFFFEH	Processor mode control register	PMC		R/W		√	√	–	00H	√	√	√	√

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

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

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

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

Extended SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe 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. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating the address of an extended SFR. It is a reserved word in the 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 can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

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

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

Table 3-6. Extended SFR (2nd SFR) List (1/13)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
					1-bit	8-bit	16-bit		48	64	80	100
F0017H	A/D port configuration register	ADPC		R/W	–	√	–	00H[Note]	√	√	√	√
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	–	√	√	√
F0036H	Pull-up resistor option register 6	PU6		R/W	√	√	–	00H	√	√	√	√
F0037H	Pull-up resistor option register 7	PU7		R/W	√	√	–	00H	√	√	√	√
F003CH	Pull-up resistor option register 12	PU12		R/W	√	√	–	00H	√	√	√	√
F003EH	Pull-up resistor option register 14	PU14		R/W	√	√	–	00H	√	√	√	√
F003FH	Pull-up resistor option register 15	PU15		R/W	√	√	–	00H	–	–	–	√
F0046H	Port input mode register 6	PIM6		R/W	√	√	–	00H	√	√	√	√
F0047H	Port input mode register 7	PIM7		R/W	√	√	–	00H	√	√	√	√
F0054H	Port output mode register 4	POM4		R/W	√	√	–	00H	–	√	√	√
F0057H	Port output mode register 7	POM7		R/W	√	√	–	00H	√	√	√	√
F0060H	Noise filter enable register 0	NFEN0		R/W	√	√	–	00H	√	√	√	√
F0061H	Noise filter enable register 1	NFEN1		R/W	√	√	–	00H	√	√	√	√
F0062H	Noise filter enable register 2	NFEN2		R/W	√	√	–	00H	√	√	√	√
F0063H	Noise filter enable register 3	NFEN3		R/W	√	√	–	00H	–	√	√	√
F0067H	Low-voltage detection flag output enable register	LVIOUT		R/W	√	√	–	00H	√	√	√	√
F006FH	Port output slew rate select register	PSRSEL		R/W	√	√	–	00H	√	√	√	√
F0070H	Specific register manipulation protection register	GUARD		R/W	√	√	–	00H	√	√	√	√
F0071H	Register used to change the operating mode of the watchdog timer during self programming	WDTSELF		R/W	–	√	–	00H	√	√	√	√
F0074H	Illegal-memory access detection control register	IAWCTL		R/W	√	√	–	00H	√	√	√	√
F0075H	Illegal-memory access RAM size setting register	IAWRAM		R/W	–	√	–	00H	√	√	√	√
F0076H	Illegal-memory access FLASH size setting register	IAWFLASH		R/W	–	√	–	00H	√	√	√	√
F0080H	DMA SFR address register 2	DSA2		R/W	–	√	–	00H	√	√	√	√
F0081H	DMA SFR address register 3	DSA3		R/W	–	√	–	00H	√	√	√	√
F0082H	DMA RAM address register 0L	DRA2L	DRA2	R/W	–	√	√	00H	√	√	√	√
F0083H	DMA RAM address register 0H	DRA2H		R/W	–	√		00H	√	√	√	√
F0084H	DMA RAM address register 1L	DRA3L	DRA3	R/W	–	√	√	00H	√	√	√	√
F0085H	DMA RAM address register 1H	DRA3H		R/W	–	√		00H	√	√	√	√
F0086H	DMA byte count register 2L	DBC2L	DBC2	R/W	–	√	√	00H	√	√	√	√
F0087H	DMA byte count register 2H	DBC2H		R/W	–	√		00H	√	√	√	√
F0088H	DMA byte count register 3L	DBC3L	DBC3	R/W	–	√	√	00H	√	√	√	√
F0089H	DMA byte count register 3H	DBC3H		R/W	–	√		00H	√	√	√	√

Note The ADPC register is not reset even if PER0.ADCEN = 0 is set.

Table 3-6. Extended SFR (2nd SFR) List (2/13)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
					1-bit	8-bit	16-bit		48	64	80	100
F008AH	DMA mode control register 2	DMC2		R/W	√	√	–	00H	√	√	√	√
F008BH	DMA mode control register 3	DMC3		R/W	√	√	–	00H	√	√	√	√
F008CH	DMA operation control register 2	DRC2		R/W	√	√	–	00H	√	√	√	√
F008DH	DMA operation control register 3	DRC3		R/W	√	√	–	00H	√	√	√	√
F008FH	DMA all-channel forced wait register	DMCALL		R/W	√	√	–	00H	√	√	√	√
F00E0H	Multiplication/division data registers C	MDCL		R	–	–	√	0000H	√	√	√	√
F00E1H												
F00E2H		MDCH		R	–	–	√	0000H	√	√	√	√
F00E3H												
F00E8H	Multiplication/division control register	MDUC		R/W	√	√	–	00H	√	√	√	√
F00F0H	Peripheral enable register 0	PER0		R/W	√	√	–	00H	√	√	√	√
F00F1H	Peripheral enable register 1	PER1		R/W	√	√	–	00H	√	√	√	√
F00F2H	Peripheral clock select register	PCKSEL		R/W	√	√	–	00H	√	√	√	√
F00F3H	Operation speed mode control register	OSMC		R/W	√	√	–	00H	√	√	√	√
F00F6H	PLL status register	PLLSTS		R	√	√	–	00H	√	√	√	√
F00F7H	PLL control register	PLLCTL		R/W	√	√	–	00H	√	√	√	√
F00FBH	POC reset register	POCRES		R/W	√	√	–	00H	√	√	√	√
F00FCH	STOP status output control register	STPSTC		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 flag clear trigger register 00	SIR00L	SIR00	R/W	–	√	√	0000H	√	√	√	√
F0105H		–			–	–						
F0106H	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	–	√	√	0000H	–	√	√	√
F0107H		–			–	–						
F0108H	Serial mode register 00	SMR00		R/W	–	–	√	0020H	√	√	√	√
F0109H												
F010AH	Serial mode register 01	SMR01		R/W	–	–	√	0020H	–	√	√	√
F010BH												
F010CH	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H	√	√	√	√
F010DH												
F010EH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H	–	√	√	√
F010FH												

Table 3-6. Extended SFR (2nd SFR) List (3/13)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
					1-bit	8-bit	16-bit		48	64	80	100
F0110H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H	√	√	√	√
F0111H		–			–	–						
F0112H	Serial channel start trigger register 0	SS0L	SS0	R/W	√	√	√	0000H	√	√	√	√
F0113H		–			–	–						
F0114H	Serial channel stop trigger register 0	ST0L	ST0	R/W	√	√	√	0000H	√	√	√	√
F0115H		–			–	–						
F0116H	Serial clock select register 0	SPS0L	SPS0	R/W	–	√	√	0000H	√	√	√	√
F0117H		–			–	–						
F0118H	Serial output register 0	SO0		R/W	–	–	√	0303H	√	√	√	√
F0119H												
F011AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H	√	√	√	√
F011BH		–			–	–						
F0120H	Serial output level register 0	SOL0L	SOL0	R/W	–	√	√	0000H	√	√	√	√
F0121H		–			–	–						
F0122H	Serial slave select enable register 0	SSE0L	SSE0	R/W	√	√	√	0000H	√	√	√	√
F0123H		–			–	–						
F0130H	Serial status register 10	SSR10L	SSR10	R	–	√	√	0000H	√	√	√	√
F0131H		–			–	–						
F0132H	Serial status register 11	SSR11L	SSR11	R	–	√	√	0000H	√	√	√	√
F0133H		–			–	–						
F0134H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	–	√	√	0000H	√	√	√	√
F0135H		–			–	–						
F0136H	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	–	√	√	0000H	√	√	√	√
F0137H		–			–	–						
F0138H	Serial mode register 10	SMR10		R/W	–	–	√	0020H	√	√	√	√
F0139H												
F013AH	Serial mode register 11	SMR11		R/W	–	–	√	0020H	√	√	√	√
F013BH												
F013CH	Serial communication operation setting register 10	SCR10		R/W	–	–	√	0087H	√	√	√	√
F013DH												
F013EH	Serial communication operation setting register 11	SCR11		R/W	–	–	√	0087H	√	√	√	√
F013FH												

Table 3-6. Extended SFR (2nd SFR) List (4/13)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
					1-bit	8-bit	16-bit		48	64	80	100
F0140H	Serial channel enable status register 1	SE1L	SE1	R	√	√	√	0000H	√	√	√	√
F0141H		–			–	–						
F0142H	Serial channel start trigger register 1	SS1L	SS1	R/W	√	√	√	0000H	√	√	√	√
F0143H		–			–	–						
F0144H	Serial channel stop trigger register 1	ST1L	ST1	R/W	√	√	√	0000H	√	√	√	√
F0145H		–			–	–						
F0146H	Serial clock select register 1	SPS1L	SPS1	R/W	–	√	√	0000H	√	√	√	√
F0147H		–			–	–						
F0148H	Serial output register 1	SO1		R/W	–	–	√	0303H	√	√	√	√
F0149H												
F014AH	Serial output enable register 1	SOE1L	SOE1	R/W	√	√	√	0000H	√	√	√	√
F014BH		–			–	–						
F0150H	Serial output level register 1	SOL1L	SOL1	R/W	–	√	√	0000H	√	√	√	√
F0151H		–			–	–						
F0158H	Serial output level register 2	SOL2L	SOL2	R/W	–	√	√	0000H	–	√	√	√
F0159H		–			–	–						
F0160H	Serial status register 20	SSR20L	SSR20	R	–	√	√	0000H	–	√	√	√
F0161H		–			–	–						
F0162H	Serial status register 21	SSR21L	SSR21	R	–	√	√	0000H	–	√	√	√
F0163H		–			–	–						
F0164H	Serial flag clear trigger register 20	SIR20L	SIR20	R/W	–	√	√	0000H	–	√	√	√
F0165H		–			–	–						
F0166H	Serial flag clear trigger register 21	SIR21L	SIR21	R/W	–	√	√	0000H	–	√	√	√
F0167H		–			–	–						
F0168H	Serial mode register 20	SMR20		R/W	–	–	√	0020H	–	√	√	√
F0169H												
F016AH	Serial mode register 21	SMR21		R/W	–	–	√	0020H	–	√	√	√
F016BH												
F016CH	Serial communication operation setting register 20	SCR20		R/W	–	–	√	0087H	–	√	√	√
F016DH												
F016EH	Serial communication operation setting register 21	SCR21		R/W	–	–	√	0087H	–	√	√	√
F016FH												
F0170H	Serial channel enable status register 2	SE2L	SE2	R	√	√	√	0000H	–	√	√	√
F0171H		–			–	–						
F0172H	Serial channel start trigger register 2	SS2L	SS2	R/W	√	√	√	0000H	–	√	√	√
F0173H		–			–	–						
F0174H	Serial channel stop trigger register 2	ST2L	ST2	R/W	√	√	√	0000H	–	√	√	√
F0175H		–			–	–						
F0176H	Serial clock select register 2	SPS2L	SPS2	R/W	–	√	√	0000H	–	√	√	√
F0177H		–			–	–						
F0178H	Serial output register 2	SO2		R/W	–	–	√	0303H	–	√	√	√
F0179H												
F017AH	Serial output enable register 2	SOE2L	SOE2	R/W	√	√	√	0000H	–	√	√	√
F017BH		–			–	–						

Table 3-6. Extended SFR (2nd SFR) List (5/13)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
					1-bit	8-bit	16-bit		48	64	80	100
F0180H	Timer counter register 00	TCR00		R	–	–	√	FFFFH	√	√	√	√
F0181H												
F0182H	Timer counter register 01	TCR01		R	–	–	√	FFFFH	√	√	√	√
F0183H												
F0184H	Timer counter register 02	TCR02		R	–	–	√	FFFFH	√	√	√	√
F0185H												
F0186H	Timer counter register 03	TCR03		R	–	–	√	FFFFH	√	√	√	√
F0187H												
F0188H	Timer counter register 04	TCR04		R	–	–	√	FFFFH	√	√	√	√
F0189H												
F018AH	Timer counter register 05	TCR05		R	–	–	√	FFFFH	√	√	√	√
F018BH												
F018CH	Timer counter register 06	TCR06		R	–	–	√	FFFFH	√	√	√	√
F018DH												
F018EH	Timer counter register 07	TCR07		R	–	–	√	FFFFH	√	√	√	√
F018FH												
F0190H	Timer mode register 00	TMR00		R/W	–	–	√	0000H	√	√	√	√
F0191H												
F0192H	Timer mode register 01	TMR01		R/W	–	–	√	0000H	√	√	√	√
F0193H												
F0194H	Timer mode register 02	TMR02		R/W	–	–	√	0000H	√	√	√	√
F0195H												
F0196H	Timer mode register 03	TMR03		R/W	–	–	√	0000H	√	√	√	√
F0197H												
F0198H	Timer mode register 04	TMR04		R/W	–	–	√	0000H	√	√	√	√
F0199H												
F019AH	Timer mode register 05	TMR05		R/W	–	–	√	0000H	√	√	√	√
F019BH												
F019CH	Timer mode register 06	TMR06		R/W	–	–	√	0000H	√	√	√	√
F019DH												
F019EH	Timer mode register 07	TMR07		R/W	–	–	√	0000H	√	√	√	√
F019FH												
F01A0H	Timer status register 00	TSR00L	TSR00	R	–	√	√	0000H	√	√	√	√
F01A1H		–			–	–						
F01A2H	Timer status register 01	TSR01L	TSR01	R	–	√	√	0000H	√	√	√	√
F01A3H		–			–	–						
F01A4H	Timer status register 02	TSR02L	TSR02	R	–	√	√	0000H	√	√	√	√
F01A5H		–			–	–						
F01A6H	Timer status register 03	TSR03L	TSR03	R	–	√	√	0000H	√	√	√	√
F01A7H		–			–	–						
F01A8H	Timer status register 04	TSR04L	TSR04	R	–	√	√	0000H	√	√	√	√
F01A9H		–			–	–						
F01AAH	Timer status register 05	TSR05L	TSR05	R	–	√	√	0000H	√	√	√	√
F01ABH		–			–	–						
F01ACH	Timer status register 06	TSR06L	TSR06	R	–	√	√	0000H	√	√	√	√
F01ADH		–			–	–						

Table 3-6. Extended SFR (2nd SFR) List (6/13)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
					1-bit	8-bit	16-bit		48	64	80	100
F01AEH	Timer status register 07	TSR07L	TSR07	R	–	√	√	0000H	√	√	√	√
F01AFH		–			–	–						
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H	√	√	√	√
F01B1H		–			–	–						
F01B2H	Timer channel start trigger register 0	TS0L	TS0	R/W	√	√	√	0000H	√	√	√	√
F01B3H		–			–	–						
F01B4H	Timer channel stop trigger register 0	TT0L	TT0	R/W	√	√	√	0000H	√	√	√	√
F01B5H		–			–	–						
F01B6H	Timer clock select register 0	TPS0		R/W	–	–	√	0000H	√	√	√	√
F01B7H												
F01B8H	Timer output register 0	TO0L	TO0	R/W	–	√	√	0000H	√	√	√	√
F01B9H		–			–	–						
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	√	√	√	0000H	√	√	√	√
F01BBH		–			–	–						
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	–	√	√	0000H	√	√	√	√
F01BDH		–			–	–						
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	–	√	√	0000H	√	√	√	√
F01BFH		–			–	–						
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 counter register 14	TCR14		R	–	–	√	FFFFH	√	√	√	√
F01C9H												
F01CAH	Timer counter register 15	TCR15		R	–	–	√	FFFFH	√	√	√	√
F01CBH												
F01CCH	Timer counter register 16	TCR16		R	–	–	√	FFFFH	√	√	√	√
F01CDH												
F01CEH	Timer counter register 17	TCR17		R	–	–	√	FFFFH	√	√	√	√
F01CFH												
F01D0H	Timer mode register 10	TMR10		R/W	–	–	√	0000H	√	√	√	√
F01D1H												
F01D2H	Timer mode register 11	TMR11		R/W	–	–	√	0000H	√	√	√	√
F01D3H												
F01D4H	Timer mode register 12	TMR12		R/W	–	–	√	0000H	√	√	√	√
F01D5H												
F01D6H	Timer mode register 13	TMR13		R/W	–	–	√	0000H	√	√	√	√
F01D7H												
F01D8H	Timer mode register 14	TMR14		R/W	–	–	√	0000H	√	√	√	√
F01D9H												
F01DAH	Timer mode register 15	TMR15		R/W	–	–	√	0000H	√	√	√	√
F01DBH												

Table 3-6. Extended SFR (2nd SFR) List (7/13)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
					1-bit	8-bit	16-bit		48	64	80	100
F01DCH	Timer mode register 16	TMR16		R/W	–	–	√	0000H	√	√	√	√
F01DDH												
F01DEH	Timer mode register 17	TMR17		R/W	–	–	√	0000H	√	√	√	√
F01DFH												
F01E0H	Timer status register 10	TSR10L	TSR10	R	–	√	√	0000H	√	√	√	√
F01E1H		–			–	–						
F01E2H	Timer status register 11	TSR11L	TSR11	R	–	√	√	0000H	√	√	√	√
F01E3H		–			–	–						
F01E4H	Timer status register 12	TSR12L	TSR12	R	–	√	√	0000H	√	√	√	√
F01E5H		–			–	–						
F01E6H	Timer status register 13	TSR13L	TSR13	R	–	√	√	0000H	√	√	√	√
F01E7H		–			–	–						
F01E8H	Timer status register 14	TSR14L	TSR14	R	–	√	√	0000H	√	√	√	√
F01E9H		–			–	–						
F01EAH	Timer status register 15	TSR15L	TSR15	R	–	√	√	0000H	√	√	√	√
F01EBH		–			–	–						
F01ECH	Timer status register 16	TSR16L	TSR16	R	–	√	√	0000H	√	√	√	√
F01EDH		–			–	–						
F01EEH	Timer status register 17	TSR17L	TSR17	R	–	√	√	0000H	√	√	√	√
F01EFH		–			–	–						
F01F0H	Timer channel enable status register 1	TE1L	TE1	R	√	√	√	0000H	√	√	√	√
F01F1H		–			–	–						
F01F2H	Timer channel start trigger register 1	TS1L	TS1	R/W	√	√	√	0000H	√	√	√	√
F01F3H		–			–	–						
F01F4H	Timer channel stop trigger register 1	TT1L	TT1	R/W	√	√	√	0000H	√	√	√	√
F01F5H		–			–	–						
F01F6H	Timer clock select register 1	TPS1		R/W	–	–	√	0000H	√	√	√	√
F01F7H												
F01F8H	Timer output register 1	TO1L	TO1	R/W	–	√	√	0000H	√	√	√	√
F01F9H		–			–	–						
F01FAH	Timer output enable register 1	TOE1L	TOE1	R/W	√	√	√	0000H	√	√	√	√
F01FBH		–			–	–						
F01FCH	Timer output level register 1	TOL1L	TOL1	R/W	–	√	√	0000H	√	√	√	√
F01FDH		–			–	–						
F01FEH	Timer output mode register 1	TOM1L	TOM1	R/W	–	√	√	0000H	√	√	√	√
F01FFH		–			–	–						
F0200H	Timer counter register 20	TCR20		R	–	–	√	FFFFH	–	√	√	√
F0201H												
F0202H	Timer counter register 21	TCR21		R	–	–	√	FFFFH	–	√	√	√
F0203H												
F0204H	Timer counter register 22	TCR22		R	–	–	√	FFFFH	–	√	√	√
F0205H												
F0206H	Timer counter register 23	TCR23		R	–	–	√	FFFFH	–	√	√	√
F0207H												

Table 3-6. Extended SFR (2nd SFR) List (8/13)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
					1-bit	8-bit	16-bit		48	64	80	100
F0208H	Timer counter register 24	TCR24		R	–	–	√	FFFFH	–	–	–	√
F0209H												
F020AH	Timer counter register 25	TCR25		R	–	–	√	FFFFH	–	–	–	√
F020BH												
F020CH	Timer counter register 26	TCR26		R	–	–	√	FFFFH	–	–	–	√
F020DH												
F020EH	Timer counter register 27	TCR27		R	–	–	√	FFFFH	–	–	–	√
F020FH												
F0210H	Timer mode register 20	TMR20		R/W	–	–	√	0000H	–	√	√	√
F0211H												
F0212H	Timer mode register 21	TMR21		R/W	–	–	√	0000H	–	√	√	√
F0213H												
F0214H	Timer mode register 22	TMR22		R/W	–	–	√	0000H	–	√	√	√
F0215H												
F0216H	Timer mode register 23	TMR23		R/W	–	–	√	0000H	–	√	√	√
F0217H												
F0218H	Timer mode register 24	TMR24		R/W	–	–	√	0000H	–	–	–	√
F0219H												
F021AH	Timer mode register 25	TMR25		R/W	–	–	√	0000H	–	–	–	√
F021BH												
F021CH	Timer mode register 26	TMR26		R/W	–	–	√	0000H	–	–	–	√
F021DH												
F021EH	Timer mode register 27	TMR27		R/W	–	–	√	0000H	–	–	–	√
F021FH												
F0220H	Timer status register 20	TSR20L	TSR20	R	–	√	√	0000H	–	√	√	√
F0221H		–			–	–						
F0222H	Timer status register 21	TSR21L	TSR21	R	–	√	√	0000H	–	√	√	√
F0223H		–			–	–						
F0224H	Timer status register 22	TSR22L	TSR22	R	–	√	√	0000H	–	√	√	√
F0225H		–			–	–						
F0226H	Timer status register 23	TSR23L	TSR23	R	–	√	√	0000H	–	√	√	√
F0227H		–			–	–						
F0228H	Timer status register 24	TSR24L	TSR24	R	–	√	√	0000H	–	–	–	√
F0229H		–			–	–						
F022AH	Timer status register 25	TSR25L	TSR25	R	–	√	√	0000H	–	–	–	√
F022BH		–			–	–						
F022CH	Timer status register 26	TSR26L	TSR26	R	–	√	√	0000H	–	–	–	√
F022DH		–			–	–						
F022EH	Timer status register 27	TSR27L	TSR27	R	–	√	√	0000H	–	–	–	√
F022FH		–			–	–						

Table 3-6. Extended SFR (2nd SFR) List (9/13)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
					1-bit	8-bit	16-bit		48	64	80	100
F0230H	Timer channel enable status register 2	TE2L	TE2	R	√	√	√	0000H	–	√	√	√
F0231H		–			–	–						
F0232H	Timer channel start trigger register 2	TS2L	TS2	R/W	√	√	√	0000H	–	√	√	√
F0233H		–			–	–						
F0234H	Timer channel stop trigger register 2	TT2L	TT2	R/W	√	√	√	0000H	–	√	√	√
F0235H		–			–	–						
F0236H	Timer clock select register 2	TPS2		R/W	–	–	√	0000H	–	√	√	√
F0237H												
F0238H	Timer output register 2	TO2L	TO2	R/W	–	√	√	0000H	–	√	√	√
F0239H		–			–	–						
F023AH	Timer output enable register 2	TOE2L	TOE2	R/W	√	√	√	0000H	–	√	√	√
F023BH		–			–	–						
F023CH	Timer output level register 2	TOL2L	TOL2	R/W	–	√	√	0000H	–	√	√	√
F023DH		–			–	–						
F023EH	Timer output mode register 2	TOM2L	TOM2	R/W	–	√	√	0000H	–	√	√	√
F023FH		–			–	–						
F0240H	LIN-UART0 control register 0	UF0CTL0		R/W	√	√	–	10H	√	√	√	√
F0241H	LIN-UART0 option control register 0	UF0OPT0		R/W	√	√	–	14H	√	√	√	√
F0242H	LIN-UART0 control register 1	UF0CTL1		R/W	–	–	√	0FFFH	√	√	√	√
F0243H												
F0244H	LIN-UART0 option control register 1	UF0OPT1		R/W	√	√	–	00H	√	√	√	√
F0245H	LIN-UART0 option control register 2	UF0OPT2		R/W	√	√	–	00H	√	√	√	√
F0246H	LIN-UART0 status register	UF0STR		R	–	–	√	0000H	√	√	√	√
F0247H												
F0248H	LIN-UART0 status clear register	UF0STC		R/W	–	–	√	0000H	√	√	√	√
F0249H												
F024AH	LIN-UART0 8-bit wait transmit data register	UF0WTXB		W	–	√	–	00H	√	√	√	√
F024BH	LIN-UART0 wait transmit data register	UF0WTX		W	–	–	√	0000H	√	√	√	√
F024EH	LIN-UART0 ID setting register	UF0ID		R/W	–	√	–	00H	√	√	√	√
F024FH	LIN-UART0 buffer register 0	UF0BUF0		R/W	–	√	–	00H	√	√	√	√
F0250H	LIN-UART0 buffer register 1	UF0BUF1		R/W	–	√	–	00H	√	√	√	√
F0251H	LIN-UART0 buffer register 2	UF0BUF2		R/W	–	√	–	00H	√	√	√	√
F0252H	LIN-UART0 buffer register 3	UF0BUF3		R/W	–	√	–	00H	√	√	√	√
F0253H	LIN-UART0 buffer register 4	UF0BUF4		R/W	–	√	–	00H	√	√	√	√
F0254H	LIN-UART0 buffer register 5	UF0BUF5		R/W	–	√	–	00H	√	√	√	√
F0255H	LIN-UART0 buffer register 6	UF0BUF6		R/W	–	√	–	00H	√	√	√	√
F0256H	LIN-UART0 buffer register 7	UF0BUF7		R/W	–	√	–	00H	√	√	√	√
F0257H	LIN-UART0 buffer register 8	UF0BUF8		R/W	–	√	–	00H	√	√	√	√
F0258H	LIN-UART0 buffer control register	UF0BUCTL		R/W	–	–	√	0000H	√	√	√	√
F0259H												
F0260H	LIN-UART1 control register 0	UF1CTL0		R/W	√	√	–	10H	√	√	√	√
F0261H	LIN-UART1 option control register 0	UF1OPT0		R/W	√	√	–	14H	√	√	√	√
F0262H	LIN-UART1 control register 1	UF1CTL1		R/W	–	–	√	0FFFH	√	√	√	√
F0263H												

Table 3-6. Extended SFR (2nd SFR) List (10/13)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
				1-bit	8-bit	16-bit		48	64	80	100
F0264H	LIN-UART1 option control register 1	UF1OPT1	R/W	√	√	–	00H	√	√	√	√
F0265H	LIN-UART1 option control register 2	UF1OPT2	R/W	√	√	–	00H	√	√	√	√
F0266H	LIN-UART1 status register	UF1STR	R	–	–	√	0000H	√	√	√	√
F0267H											
F0268H	LIN-UART1 status clear register	UF1STC	R/W	–	–	√	0000H	√	√	√	√
F0269H											
F026AH	LIN-UART1 8-bit wait transmit data register	UF1WTXB	W	–	√	–	00H	√	√	√	√
F026BH	LIN-UART1 wait transmit data register	UF1WTX	W	–	–	√	0000H	√	√	√	√
F026EH	LIN-UART1 ID setting register	UF1ID	R/W	–	√	–	00H	√	√	√	√
F026FH	LIN-UART1 buffer register 0	UF1BUF1	R/W	–	√	–	00H	√	√	√	√
F0270H	LIN-UART1 buffer register 1	UF1BUF1	R/W	–	√	–	00H	√	√	√	√
F0271H	LIN-UART1 buffer register 2	UF1BUF2	R/W	–	√	–	00H	√	√	√	√
F0272H	LIN-UART1 buffer register 3	UF1BUF3	R/W	–	√	–	00H	√	√	√	√
F0273H	LIN-UART1 buffer register 4	UF1BUF4	R/W	–	√	–	00H	√	√	√	√
F0274H	LIN-UART1 buffer register 5	UF1BUF5	R/W	–	√	–	00H	√	√	√	√
F0275H	LIN-UART1 buffer register 6	UF1BUF6	R/W	–	√	–	00H	√	√	√	√
F0276H	LIN-UART1 buffer register 7	UF1BUF7	R/W	–	√	–	00H	√	√	√	√
F0277H	LIN-UART1 buffer register 8	UF1BUF8	R/W	–	√	–	00H	√	√	√	√
F0278H	LIN-UART1 buffer control register	UF1BUCTL	R/W	–	–	√	0000H	√	√	√	√
F0279H											
F0280H	10-bit A/D conversion result register 0	ADCR0	R	–	–	√	0000H	√	√	√	√
F0281H	8-bit A/D conversion result register 0	ADCR0H	R	–	√	–	00H	√	√	√	√
F0282H	10-bit A/D conversion result register 1	ADCR1	R	–	–	√	0000H	√	√	√	√
F0283H	8-bit A/D conversion result register 1	ADCR1H	R	–	√	–	00H	√	√	√	√
F0284H	10-bit A/D conversion result register 2	ADCR2	R	–	–	√	0000H	√	√	√	√
F0285H	8-bit A/D conversion result register 2	ADCR2H	R	–	√	–	00H	√	√	√	√
F0286H	10-bit A/D conversion result register 3	ADCR3	R	–	–	√	0000H	√	√	√	√
F0287H	8-bit A/D conversion result register 3	ADCR3H	R	–	√	–	00H	√	√	√	√
F0288H	10-bit A/D conversion result register 4	ADCR4	R	–	–	√	0000H	√	√	√	√
F0289H	8-bit A/D conversion result register 4	ADCR4H	R	–	√	–	00H	√	√	√	√
F028AH	10-bit A/D conversion result register 5	ADCR5	R	–	–	√	0000H	√	√	√	√
F028BH	8-bit A/D conversion result register 5	ADCR5H	R	–	√	–	00H	√	√	√	√
F028CH	10-bit A/D conversion result register 6	ADCR6	R	–	–	√	0000H	√	√	√	√
F028DH	8-bit A/D conversion result register 6	ADCR6H	R	–	√	–	00H	√	√	√	√
F028EH	10-bit A/D conversion result register 7	ADCR7	R	–	–	√	0000H	√	√	√	√
F028FH	8-bit A/D conversion result register 7	ADCR7H	R	–	√	–	00H	√	√	√	√

Table 3-6. Extended SFR (2nd SFR) List (11/13)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
				1-bit	8-bit	16-bit		48	64	80	100
F0290H	10-bit A/D conversion result register 8	ADCR8	R	–	–	√	0000H	√	√	√	√
F0291H	8-bit A/D conversion result register 8	ADCR8H	R	–	√	–	00H	√	√	√	√
F0292H	10-bit A/D conversion result register 9	ADCR9	R	–	–	√	0000H	√	√	√	√
F0293H	8-bit A/D conversion result register 9	ADCR9H	R	–	√	–	00H	√	√	√	√
F0294H	10-bit A/D conversion result register 10	ADCR10	R	–	–	√	0000H	√	√	√	√
F0295H	8-bit A/D conversion result register 10	ADCR10H	R	–	√	–	00H	√	√	√	√
F0296H	10-bit A/D conversion result register 11	ADCR11	R	–	–	√	0000H	–	√	√	√
F0297H	8-bit A/D conversion result register 11	ADCR11H	R	–	√	–	00H	–	√	√	√
F0298H	10-bit A/D conversion result register 12	ADCR12	R	–	–	√	0000H	–	√	√	√
F0299H	8-bit A/D conversion result register 12	ADCR12H	R	–	√	–	00H	–	√	√	√
F029AH	10-bit A/D conversion result register 13	ADCR13	R	–	–	√	0000H	–	√	√	√
F029BH	8-bit A/D conversion result register 13	ADCR13H	R	–	√	–	00H	–	√	√	√
F029CH	10-bit A/D conversion result register 14	ADCR14	R	–	–	√	0000H	–	√	√	√
F029DH	8-bit A/D conversion result register 14	ADCR14H	R	–	√	–	00H	–	√	√	√
F029EH	10-bit A/D conversion result register 15	ADCR15	R	–	–	√	0000H	–	–	√	√
F029FH	8-bit A/D conversion result register 15	ADCR15H	R	–	√	–	00H	–	–	√	√
F02A0H	10-bit A/D conversion result register 16	ADCR16	R	–	–	√	0000H	–	–	–	√
F02A1H	8-bit A/D conversion result register 16	ADCR16H	R	–	√	–	00H	–	–	–	√
F02A2H	10-bit A/D conversion result register 17	ADCR17	R	–	–	√	0000H	–	–	–	√
F02A3H	8-bit A/D conversion result register 17	ADCR17H	R	–	√	–	00H	–	–	–	√
F02A4H	10-bit A/D conversion result register 18	ADCR18	R	–	–	√	0000H	–	–	–	√
F02A5H	8-bit A/D conversion result register 18	ADCR18H	R	–	√	–	00H	–	–	–	√
F02A6H	10-bit A/D conversion result register 19	ADCR19	R	–	–	√	0000H	–	–	–	√
F02A7H	8-bit A/D conversion result register 19	ADCR19H	R	–	√	–	00H	–	–	–	√
F02A8H	10-bit A/D conversion result register 20	ADCR20	R	–	–	√	0000H	–	–	–	√
F02A9H	8-bit A/D conversion result register 20	ADCR20H	R	–	√	–	00H	–	–	–	√

Table 3-6. Extended SFR (2nd SFR) List (12/13)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
				1-bit	8-bit	16-bit		48	64	80	100
F02AAH	10-bit A/D conversion result register 21	ADCR21	R	–	–	√	0000H	–	–	–	√
F02ABH	8-bit A/D conversion result register 21	ADCR21H	R	–	√	–	00H	–	–	–	√
F02ACH	10-bit A/D conversion result register 22	ADCR22	R	–	–	√	0000H	–	–	–	√
F02ADH	8-bit A/D conversion result register 22	ADCR22H	R	–	√	–	00H	–	–	–	√
F02AEH	10-bit A/D conversion result register 23	ADCR23	R	–	–	√	0000H	–	–	–	√
F02AFH	8-bit A/D conversion result register 23	ADCR23H	R	–	√	–	00H	–	–	–	√
F04F0H	Data flash status register	DFLST	R	√	√	–	00H	√	√	√	√
F05C0H	CAN global module control register	CGMCTRL	R/W	–	–	√	0000H	√	√	√	√
F05C1H											
F05C6H	CAN global automatic block transmission control register	CGMABT	R/W	–	–	√	0000H	√	√	√	√
F05C7H											
F05C8H	CAN global automatic block transmission delay setting register	CGMABTD	R/W	–	√	–	00H	√	√	√	√
F05CEH	CAN global module clock select register	CGMCS	R/W	–	√	–	0FH	√	√	√	√
F05D0H	CAN module mask 1 register L	CMASK1L	R/W	–	–	√	Undefined	√	√	√	√
F05D1H											
F05D2H	CAN module mask 1 register H	CMASK1H	R/W	–	–	√	Undefined	√	√	√	√
F05D3H											
F05D4H	CAN module mask 2 register L	CMASK2L	R/W	–	–	√	Undefined	√	√	√	√
F05D5H											
F05D6H	CAN module mask 2 register H	CMASK2H	R/W	–	–	√	Undefined	√	√	√	√
F05D7H											
F05D8H	CAN module mask 3 register L	CMASK3L	R/W	–	–	√	Undefined	√	√	√	√
F05D9H											
F05DAH	CAN module mask 3 register H	CMASK3H	R/W	–	–	√	Undefined	√	√	√	√
F05DBH											
F05DCH	CAN module mask 4 register L	CMASK4L	R/W	–	–	√	Undefined	√	√	√	√
F05DDH											
F05DEH	CAN module mask 4 register H	CMASK4H	R/W	–	–	√	Undefined	√	√	√	√
F05DFH											
F05E0H	CAN module control register	CCTRL	R/W	–	–	√	0000H	√	√	√	√
F05E1H											
F05E2H	CAN module last error code register	CLEC	R/W	–	√	–	00H	√	√	√	√
F05E3H	CAN module information register	CINFO	R	–	√	–	00H	√	√	√	√
F05E4H	CAN module error counter register	CERC	R	–	–	√	0000H	√	√	√	√
F05E5H											

Table 3-6. Extended SFR (2nd SFR) List (13/13)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	HC3	HE3	HF3	HG3
				1-bit	8-bit	16-bit		48	64	80	100
F05E6H	CAN module interrupt enable register	CIE	R/W	–	–	√	0000H	√	√	√	√
F05E7H											
F05E8H	CAN module interrupt status register	CINTS	R/W	–	–	√	0000H	√	√	√	√
F05E9H											
F05EAH	CAN module bit rate prescaler register	CBRP	R/W	–	√	–	FFH	√	√	√	√
F05ECH	CAN module bit rate register	CBTR	R/W	–	–	√	370FH	√	√	√	√
F05EDH											
F05EEH	CAN module last in-pointer register	CLIPT	R/W	–	√	–	Undefined	√	√	√	√
F05F0H	CAN module receive history list register	CRGPT	R/W	–	–	√	xx02H	√	√	√	√
F05F1H											
F05F2H	CAN module last out-pointer register	CLOPT	R/W	–	√	–	Undefined	√	√	√	√
F05F4H	CAN module transmit history list register	CTGPT	R/W	–	–	√	xx02H	√	√	√	√
F05F5H											
F05F6H	CAN module time stamp register	CTS	R/W	–	–	√	0000H	√	√	√	√
F05F7H											

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

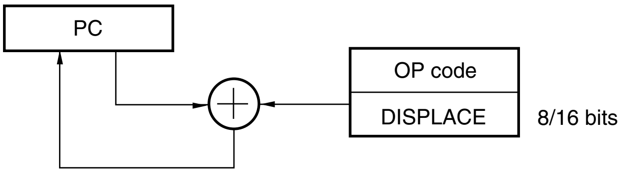
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

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

Figure 3-18. Outline of Relative Addressing



<R> **Note** Please do not use Relative addressing in the following commands.
The branch instruction from internal program memory space to RAM space.
The branch instruction from RAM space to internal program memory space.

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 3-19. Example of CALL !!addr20/BR !!addr20

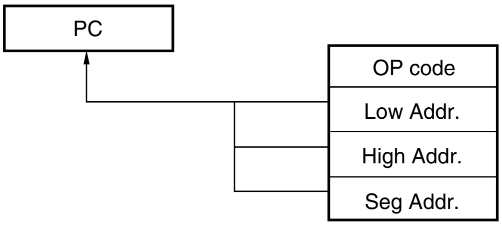
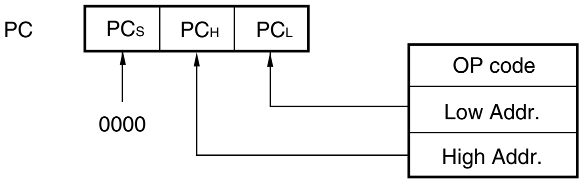


Figure 3-20. Example of CALL !addr16/BR !addr16



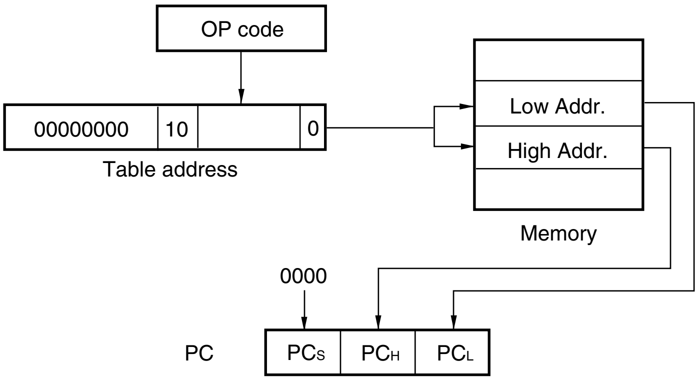
3.3.3 Table indirect addressing

[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 3-21. Outline of Table Indirect Addressing

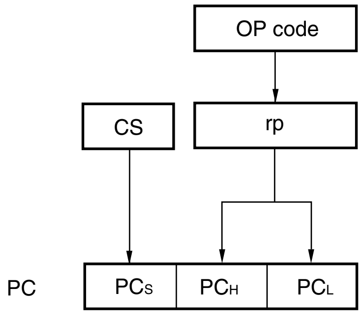


3.3.4 Register direct addressing

[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 3-22. Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

[Function]

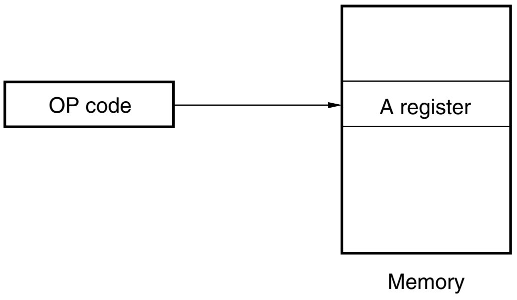
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[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 3-23. Outline of Implied Addressing



3.4.2 Register addressing

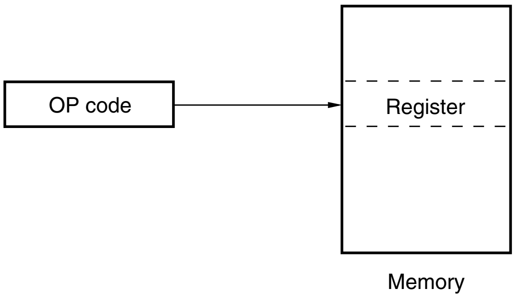
[Function]

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 3-24. 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 3-25. Example of ADDR16

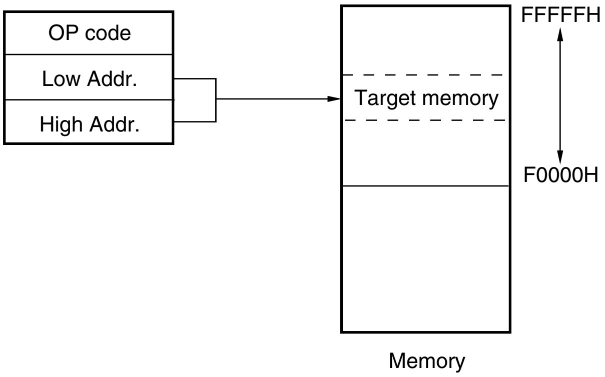
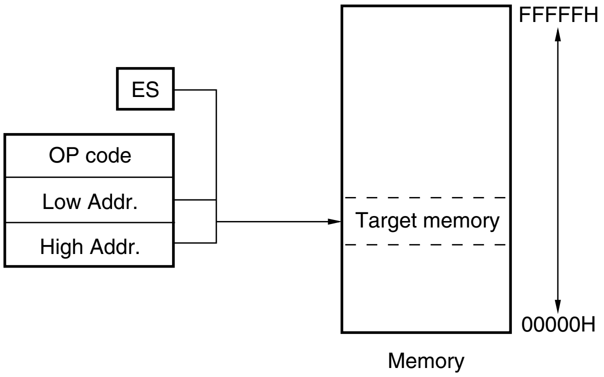


Figure 3-26. Example of ES:ADDR16



3.4.4 Short direct addressing

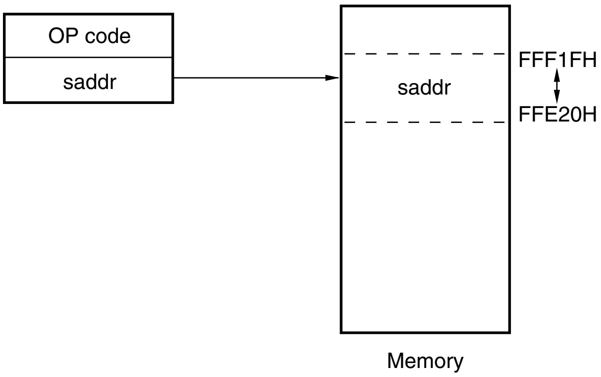
[Function]

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 3-27. 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

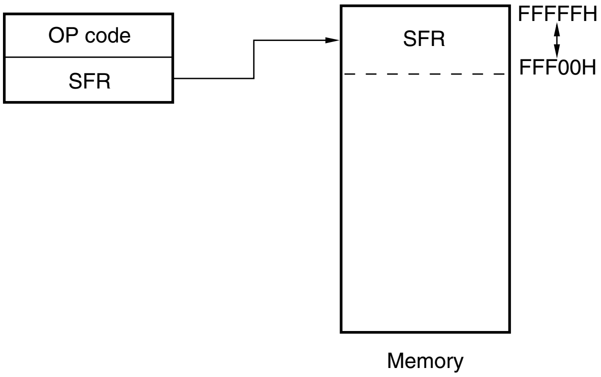
[Function]

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 3-28. 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 3-29. Example of [DE], [HL]

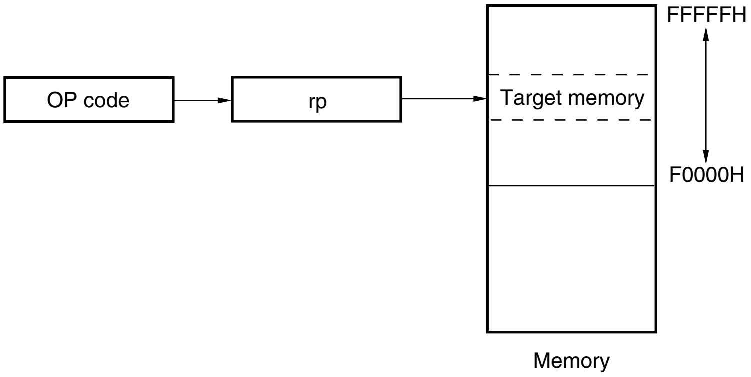
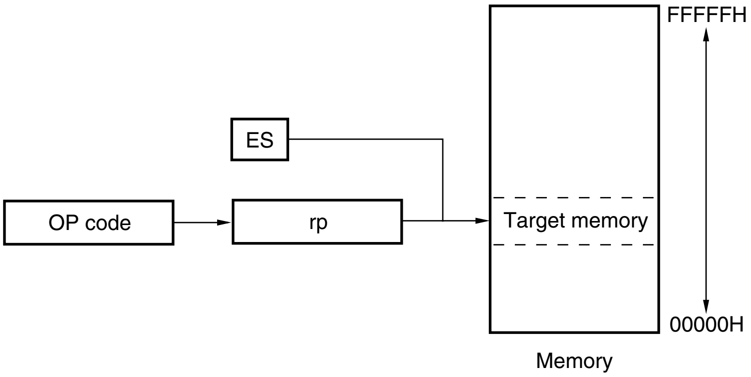


Figure 3-30. Example of ES:[DE], ES:[HL]



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)

<R> **Notes 1.** When not using ES register, it is prohibition that the result added to the base address exceeds FFFFH. When using ES register, it is prohibition that the result added to the base address exceeds FFFFFH.

<R> **2.** When using [SP + byte], SP's value being in RAM space and the value which added byte with SP need to be below FFEDFH in RAM space.

Figure 3-31. Example of [SP+byte]

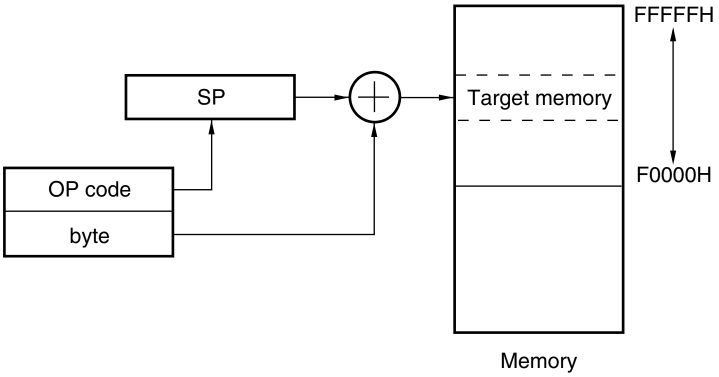


Figure 3-32. Example of [HL + byte], [DE + byte]

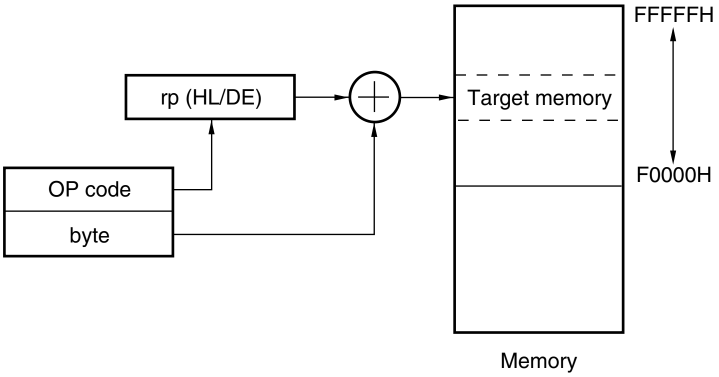


Figure 3-33. Example of word[B], word[C]

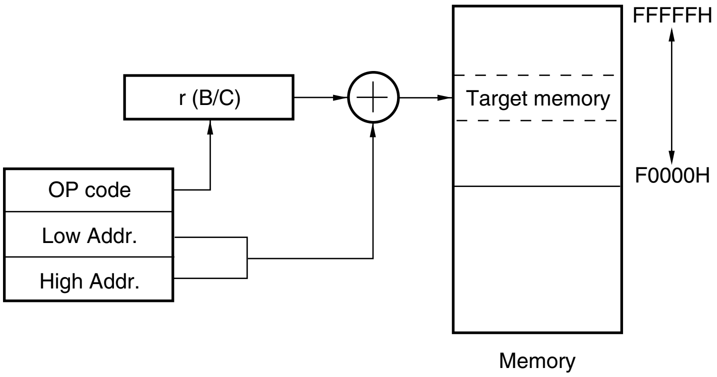


Figure 3-34. Example of word[BC]

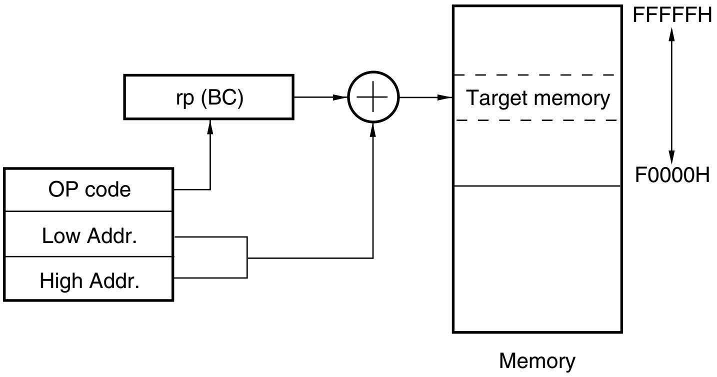


Figure 3-35. Example of ES:[HL + byte], ES:[DE + byte]

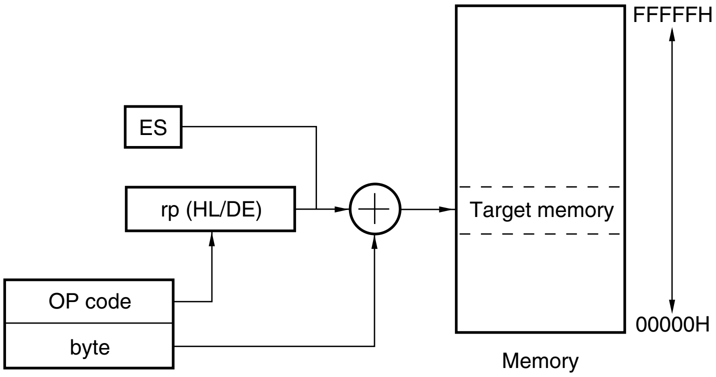


Figure 3-36. Example of ES:word[B], ES:word[C]

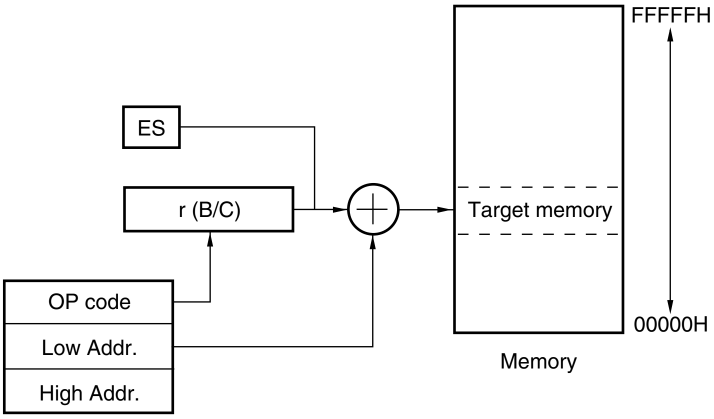
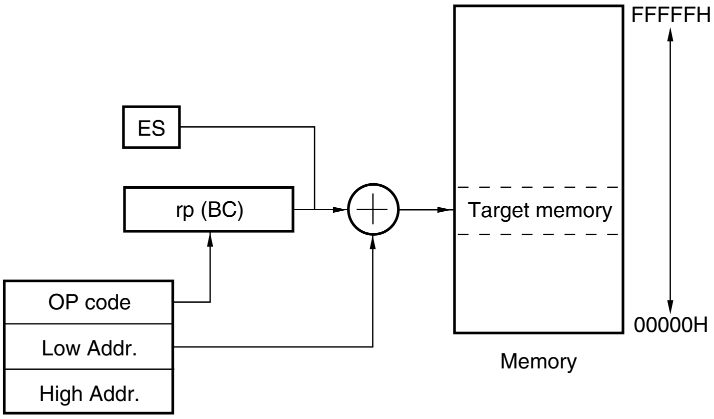


Figure 3-37. Example of ES:word[BC]



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)

<R> **Note** When not using ES register, it is prohibition that the result added to the base address exceeds FFFFH. When using ES register, it is prohibition that the result added to the base address exceeds FFFFFH.

Figure 3-38. Example of [HL+B], [HL+C]

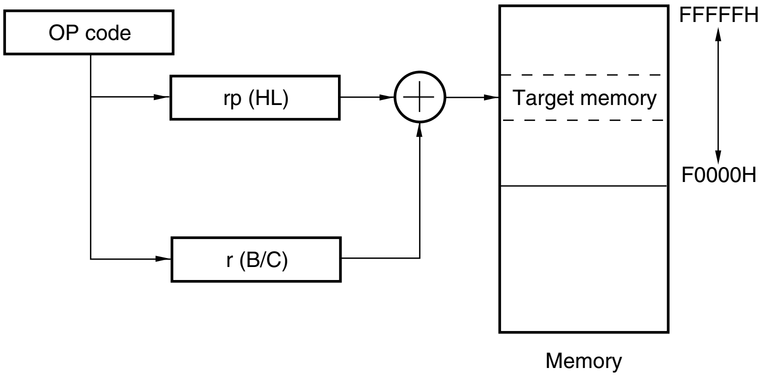
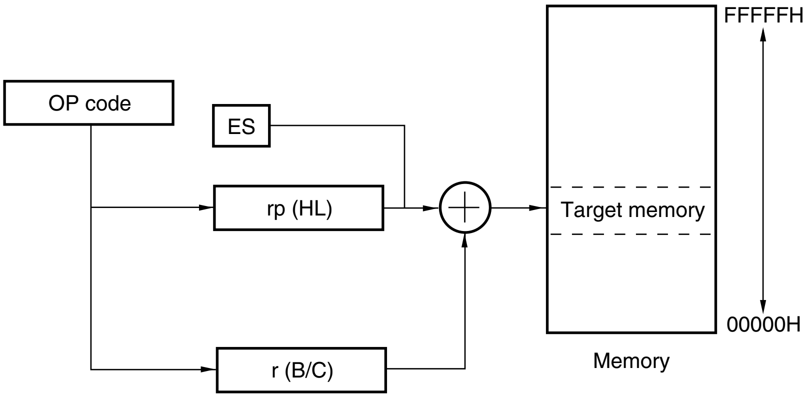


Figure 3-39. Example of ES:[HL+B], ES:[HL+C]



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 three types of pin I/O buffer power supplies: AV_{REF}, EV_{DD} (EV_{DD0}, EV_{DD1}), and V_{DD}. The relationship between these power supplies and the pins is shown below.

Table 4-1. Pin I/O Buffer Power Supplies (AV_{REF}, EV_{DD0}, V_{DD})

- 78K0R/HC3: 48-pin plastic LQFP (fine pitch) (7 × 7)
- 78K0R/HE3: 64-pin plastic LQFP (fine pitch) (10 × 10)
- 78K0R/HF3: 80-pin plastic LQFP (fine pitch) (12 × 12)

Power Supply	Corresponding Pins
AV_{REF}	P80 to P87, P90 to P97
EV_{DD}	Port pins other than P80 to P87, P90 to P97, and P121 to P124
V_{DD}	• P121 to P124 • Pins other than port pins

Table 4-2. Pin I/O Buffer Power Supplies (AV_{REF}, EV_{DD0}, EV_{DD1}, V_{DD})

- 78K0R/HG3: 100-pin plastic LQFP (fine pitch) (14 × 14)

Power Supply	Corresponding Pins
AV_{REF}	P80 to P87, P90 to P97, P100 to P107
EV_{DD0}, EV_{DD1}	Port pins other than P80 to P87, P90 to P97, P100 to P107, and P121 to P124
V_{DD}	• P121 to P124 • Pins other than port pins

78K0R/Hx3 products are provided with the digital I/O ports, which enable variety of control operations. The functions of each port are shown in Table 4-3.

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 4-3. Port Functions (1/3)

H C 3	H E 3	H F 3	H G 3	Function Name	I/O	Function	After Reset	Alternate Function
48	64	80	100					
√	√	√	√	P00	I/O	Port 0. 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	INTP7/TI05/TO05
–	–	√	√	P01				TI04/TO04
–	–	√	√	P02				TI06/TO06
–	–	–	√	P03				–
√	√	√	√	P10	I/O	Port 1. 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	TI00/SCK10/TO00/CTxD/LTxD1
√	√	√	√	P11				TI02/SI10/LRxD1/INTPLR1/CRxD/TO02
√	√	√	√	P12				INTP3/TI16/SO10/TO16
√	√	√	√	P13				TI04/LTxD0/TO04
√	√	√	√	P14				TI06/LRxD0/INTPLR0/TO06
√	√	√	√	P15				TI10/SO00/TO10
√	√	√	√	P16				TI12/SI00/TO12
√	√	√	√	P17				TI14/SCK00/TO14
√	√	√	√	P30	I/O	Port 3. 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	INTP2/SSI00/TI01/TO01
√	√	√	√	P31				INTP2/TI11/STOPST/TO11
√	√	√	√	P32				INTP4/TI13/TO13
√	√	√	√	P40	I/O	Port 4. I/O port. Output of P42 and P43 can be set to N-ch open-drain output (V$_{DD}$ tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOOL0/TI05/TO05
√	√	√	√	P41				TOOL1/TI07/TO07
–	√	√	√	P42				TxD2/SCL20
–	√	√	√	P43				RxD2/SDA20/INTPR2
–	–	√	√	P44				TI07/TO07
–	–	√	√	P45				TI10/TO10
–	–	√	√	P46				TI12/TO12
–	–	√	√	P47				INTP8

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

Table 4-3. Port Functions (2/3)

HC3	HE3	HF3	HG3	Function Name	I/O	Function	After Reset	Alternate Function
48	64	80	100					
–	√	√	√	P50	I/O	Port 5. 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	INTP3/TI20/TO20
–	√	√	√	P51				TI21/TO21
–	√	√	√	P52				TI22/STOPST/TO22
–	√	√	√	P53				TI23/TO23
–	–	√	√	P54				TI11/TO11
–	–	√	√	P55				TI13/TO13
–	–	√	√	P56				TI15/TO15
–	–	√	√	P57				TI17/TO17
√	√	√	√	P60	I/O	Port 6. I/O port. Output of P60 to P63 is N-ch open-drain output (6 V tolerance). Input of P60, P61 and P63 can be set to TTL input buffer. 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	SCK00/SCL11
√	√	√	√	P61				SI00/SDA11
√	√	√	√	P62				SO00
√	√	√	√	P63				SSI00
–	–	√	√	P64				TI14/TO14
–	–	√	√	P65				TI16/TO16
–	–	√	√	P66				TI00/TO00
–	–	√	√	P67				TI02/TO02
√	√	√	√	P70	I/O	Port 7. I/O port. Input of P73, P75 to P77 can be set to TTL input buffer. Output of P72, P74 and P76 can be set to N-ch open-drain output (VDD tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP5/KR0/TI15/TO15/ LVIOUT
√	√	√	√	P71				INTP6/KR1/TI17/TO17
√	√	√	√	P72				KR2/CTxD/LTxD1
√	√	√	√	P73				KR3/CRxD/LRxD1/ INTPLR1
–	√	√	√	P74				KR4/SO01
–	√	√	√	P75				KR5/SI01
–	√	√	√	P76				KR6/SCK01
–	√	√	√	P77				KR7/SSI01
√	√	√	√	P80	I/O	Port 8. I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI00
√	√	√	√	P81				ANI01
√	√	√	√	P82				ANI02
√	√	√	√	P83				ANI03
√	√	√	√	P84				ANI04
√	√	√	√	P85				ANI05
√	√	√	√	P86				ANI06
√	√	√	√	P87				ANI07

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

Table 4-3. Port Functions (3/3)

HC3	HE3	HF3	HG3	Function Name	I/O	Function	After Reset	Alternate Function
48	64	80	100					
√	√	√	√	P90	I/O	Port 9. I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI08
√	√	√	√	P91				ANI09
√	√	√	√	P92				ANI10
–	√	√	√	P93				ANI11
–	√	√	√	P94				ANI12
–	√	√	√	P95				ANI13
–	√	√	√	P96				ANI14
–	–	√	√	P97				ANI15
–	–	–	√	P100 to P107	I/O	Port 10. I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI16 to ANI23
√	√	√	√	P120	I/O	Port 12. I/O port and input port. Input/output can be specified in 1-bit units. For only P120 and P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI/TI11/TO11$^{\text{Note}}$
√	√	√	√	P121	Input			X1
√	√	√	√	P122				X2/EXCLK
√	√	√	√	P123				–
√	√	√	√	P124				EXCLKS
√	√	√	√	P125	I/O			INTP1/ADTRG/TI03/TO03
–	–	√	√	P126				TI01/TO01
–	–	–	√	P127				TI03/TO03
√	√	√	√	P130	Output	Port 13. Output port.	Output port	RESOUT
√	√	√	√	P140	I/O	Port 14. I/O port. Use of an on-chip pull-up resistor can be specified by a software setting.	Output port	PCL
–	–	–	√	P150	I/O	Port 15. I/O port. Input/output can be specified in 1-bit units.	Input port	–
–	–	–	√	P151				SO11
–	–	–	√	P152				SI11
–	–	–	√	P153				$\overline{\text{SCK11}}$
–	–	–	√	P154				TI24/TO24
–	–	–	√	P155				TI25/TO25
–	–	–	√	P156				TI26/TO26
–	–	–	√	P157				TI27/TO27

Note TI11, TO11 pins are 78K0R/HC3, 78K0R/HE3 only.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

4.2 Port Configuration

Ports include the following hardware.

Table 4-4. Port Configuration

Item	Configuration
Control registers	• 78K0R/HC3 Port mode register (PM0, PM1, PM3, PM4, PM6 to PM9, PM12, PM14) Port register (P0, P1, P3, P4, P6 to P9, P12 to P14) Pull-up resistor option register (PU0, PU1, PU3, PU4, PU6, PU7, PU12, PU14) Port input mode register (PIM6, PIM7) Port output mode register (POM7) A/D port configuration register (ADPC) Port output slew rate select register (PSRSEL) • 78K0R/HE3 Port mode register (PM0, PM1, PM3 to PM9, PM12, PM14) Port register (P0, P1, P3 to P9, P12 to P14) Pull-up resistor option register (PU0, PU1, PU3 to PU7, PU12, PU14) Port input mode register (PIM6, PIM7) Port output mode register (POM4, POM7) A/D port configuration register (ADPC) Port output slew rate select register (PSRSEL) • 78K0R/HF3 Port mode register (PM0, PM1, PM3 to PM9, PM12, PM14) Port register (P0, P1, P3 to P9, P12 to P14) Pull-up resistor option register (PU0, PU1, PU3 to PU7, PU12, PU14) Port input mode register (PIM6, PIM7) Port output mode register (POM4, POM7) A/D port configuration register (ADPC) Port output slew rate select register (PSRSEL) • 78K0R/HG3 Port mode register (PM0, PM1, PM3 to PM10, PM12, PM14, PM15) Port register (P0, P1, P3 to P10, P12 to P15) Pull-up resistor option register (PU0, PU1, PU3 to PU7, PU12, PU14, PU15) Port input mode register (PIM6, PIM7) Port output mode register (POM4, POM7) A/D port configuration register (ADPC) Port output slew rate select register (PSRSEL)
Port	• 78K0R/HC3: Total: 41 (CMOS I/O: 32, CMOS input: 4, CMOS output: 1, N-ch open drain I/O: 4) • 78K0R/HE3: Total: 55 (CMOS I/O: 46, CMOS input: 4, CMOS output: 1, N-ch open drain I/O: 4) • 78K0R/HF3: Total: 71 (CMOS I/O: 62, CMOS input: 4, CMOS output: 1, N-ch open drain I/O: 4) • 78K0R/HG3: Total: 89 (CMOS I/O: 80, CMOS input: 4, CMOS output: 1, N-ch open drain I/O: 4)
Pull-up resistor	• 78K0R/HC3: Total: 25 • 78K0R/HE3: Total: 35 • 78K0R/HF3: Total: 50 • 78K0R/HG3: Total: 60

4.2.1 Port 0

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P00/INTP7/TI05/TO05	√	√	√	√
P01/TI04/TO04	–	–	√	√
P02/TI06/TO06	–	–	√	√
P03	–	–	–	√

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P03 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 timer I/O, and external interrupt request input.

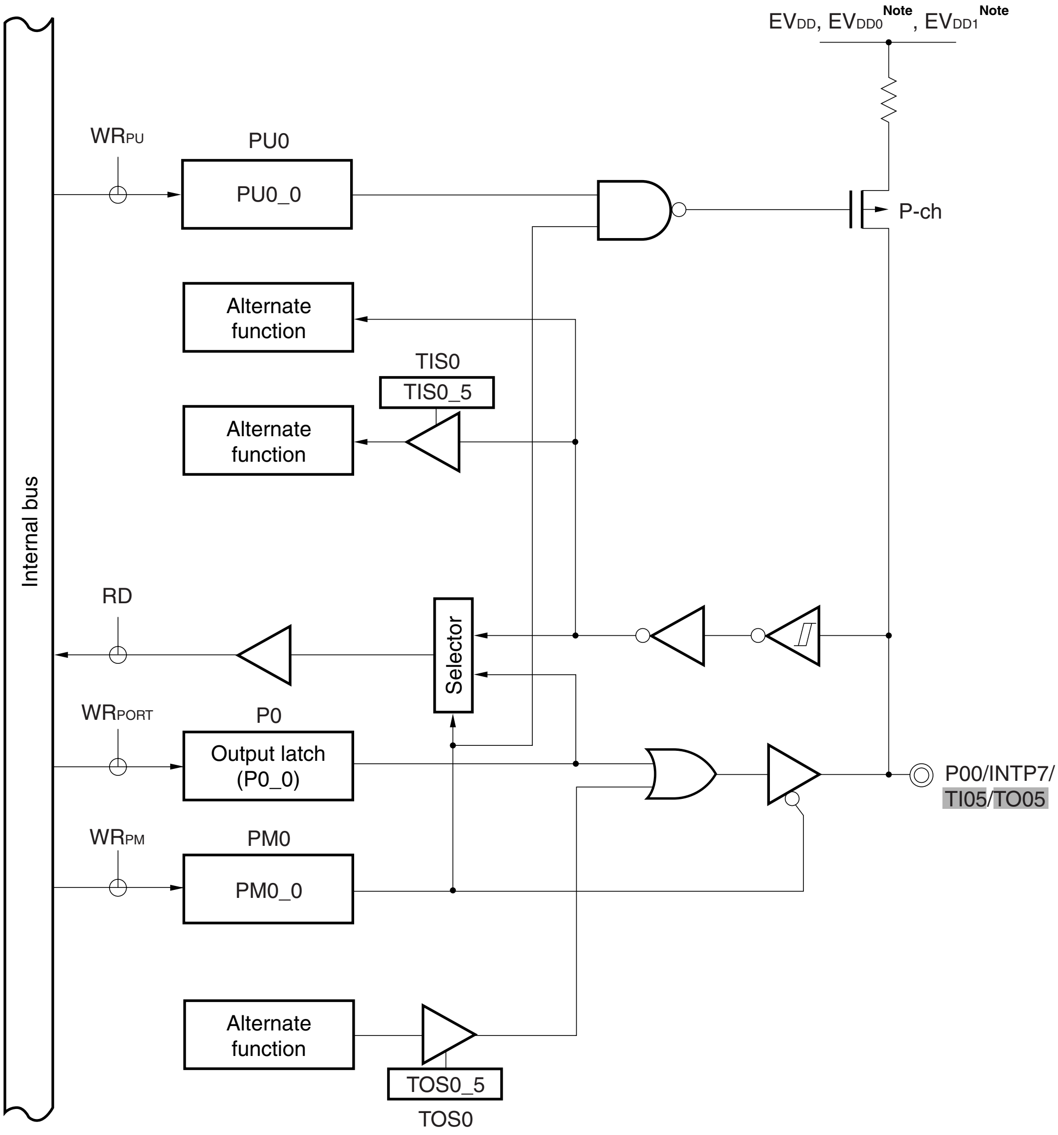
Reset signal generation sets port 0 to input mode.

Figures 4-1 to 4-3 show block diagrams of port 0.

Cautions 1. To use P00/INTP7/TI05/TO05, P01/TI04/TO04 or P02/TI06/TO06 as a general-purpose ports, configure bits 4 to 6 (TO04 to TO06) of timer output register 0 (TO0) and bits 4 to 6 (TOE0_4 to TOE0_6) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

2. The shaded pins are provided at two ports. Select either port by using the corresponding register.

Figure 4-1. Block Diagram of P00

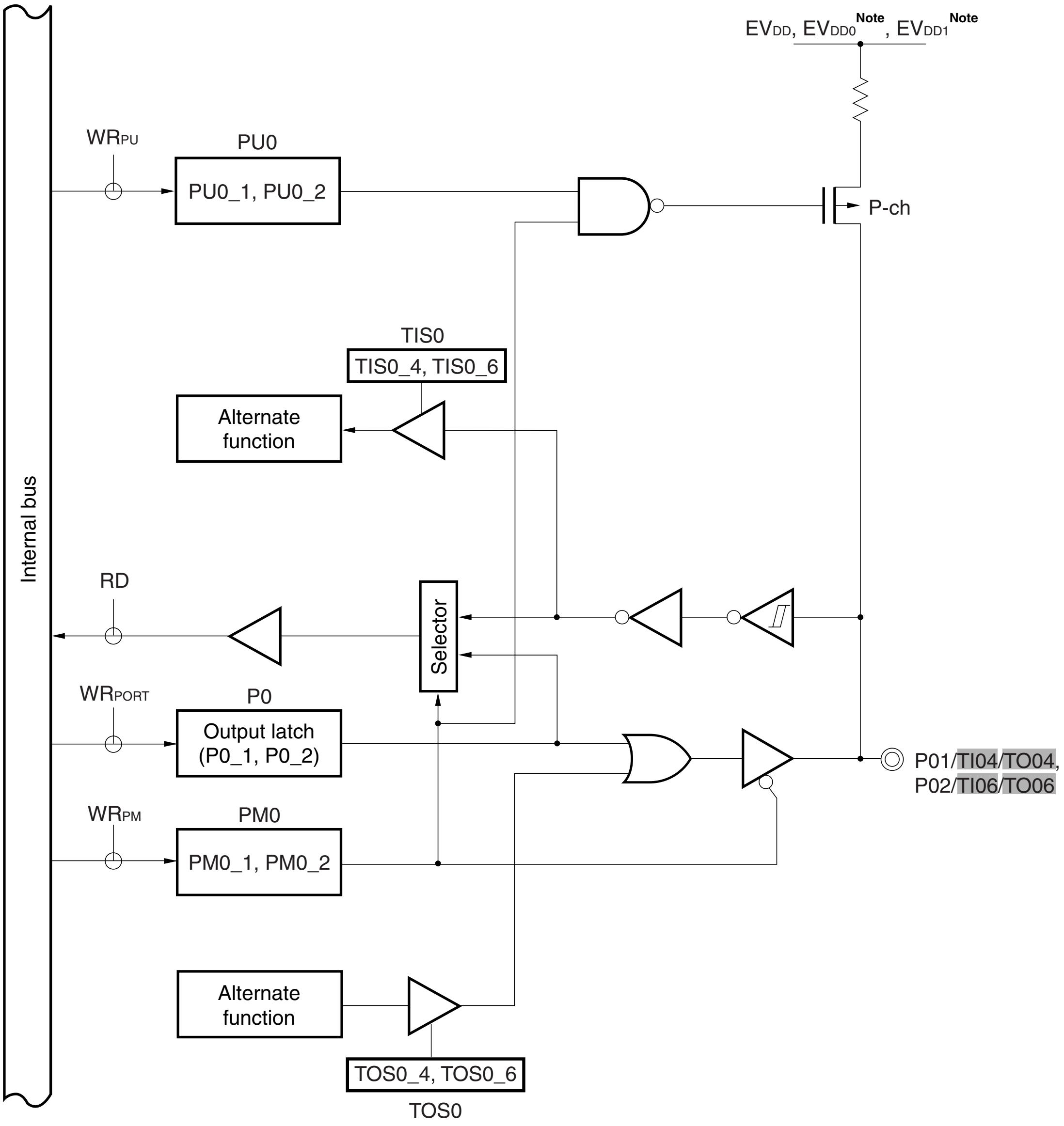


Note EVDD0, EVDD1 are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P0:	Port register 0
PU0:	Pull-up resistor option register 0
PM0:	Port mode register 0
RD:	Read signal
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

Figure 4-2. Block Diagram of P01 and P02

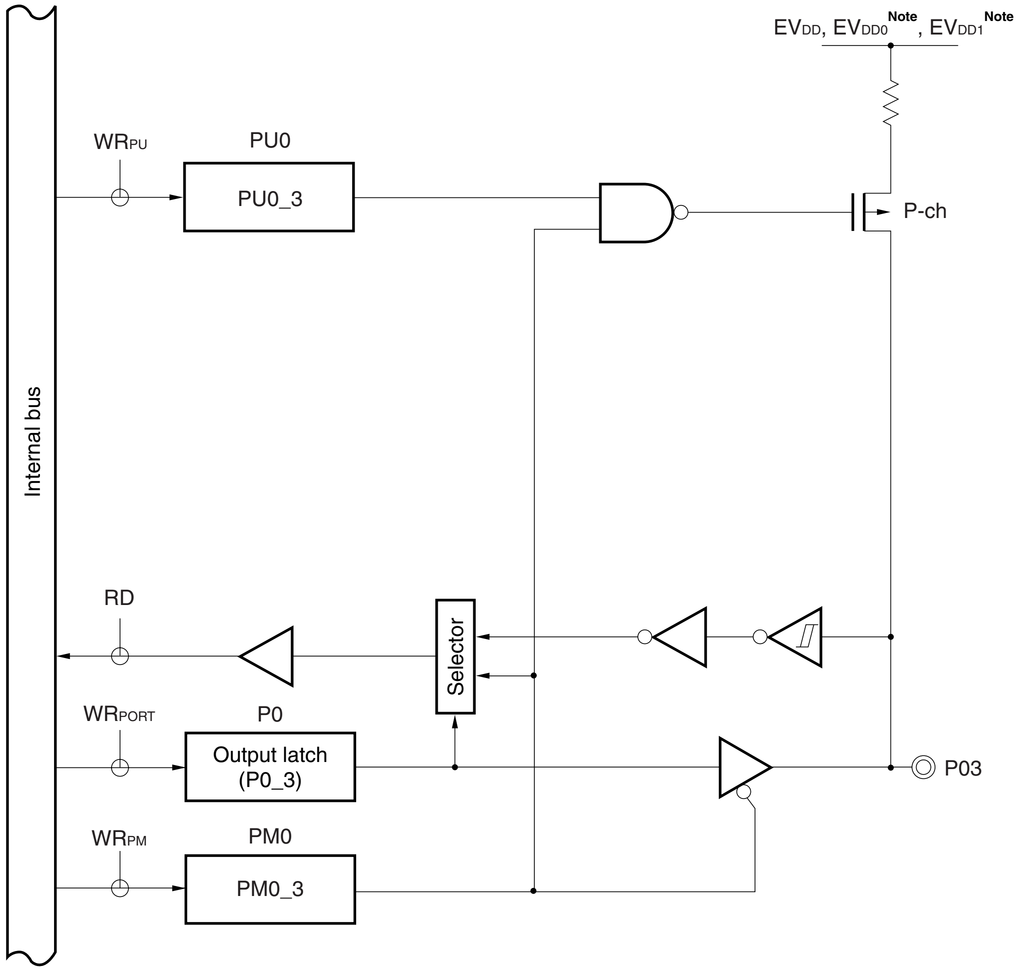


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P0:	Port register 0
PU0:	Pull-up resistor option register 0
PM0:	Port mode register 0
RD:	Read signal
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

Figure 4-3. Block Diagram of P03



Note EV_{DD0}, EV_{DD1} are 78K0R/HG3 only

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

4.2.2 Port 1

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P10//TI00/ SCK10/LTxD1/ CTxD/TO00	√			
P11/TI02/ SI10/LRxD1/CRxD/ INTPLR1/TO02	√			
P12/INTP3/TI06/ SO10/TO06	√			
P13/TI03/LTxD0/ TO03	√			
P14/TI06/LRxD0/ INTPLR0/TO06	√			
P15/TI10/SO00/ TO10	√			
P16/TI12/SI00/ TO12	√			
P17/TI14/SCK00/ TO14	√			

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

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

Reset signal generation sets port 1 to input mode.

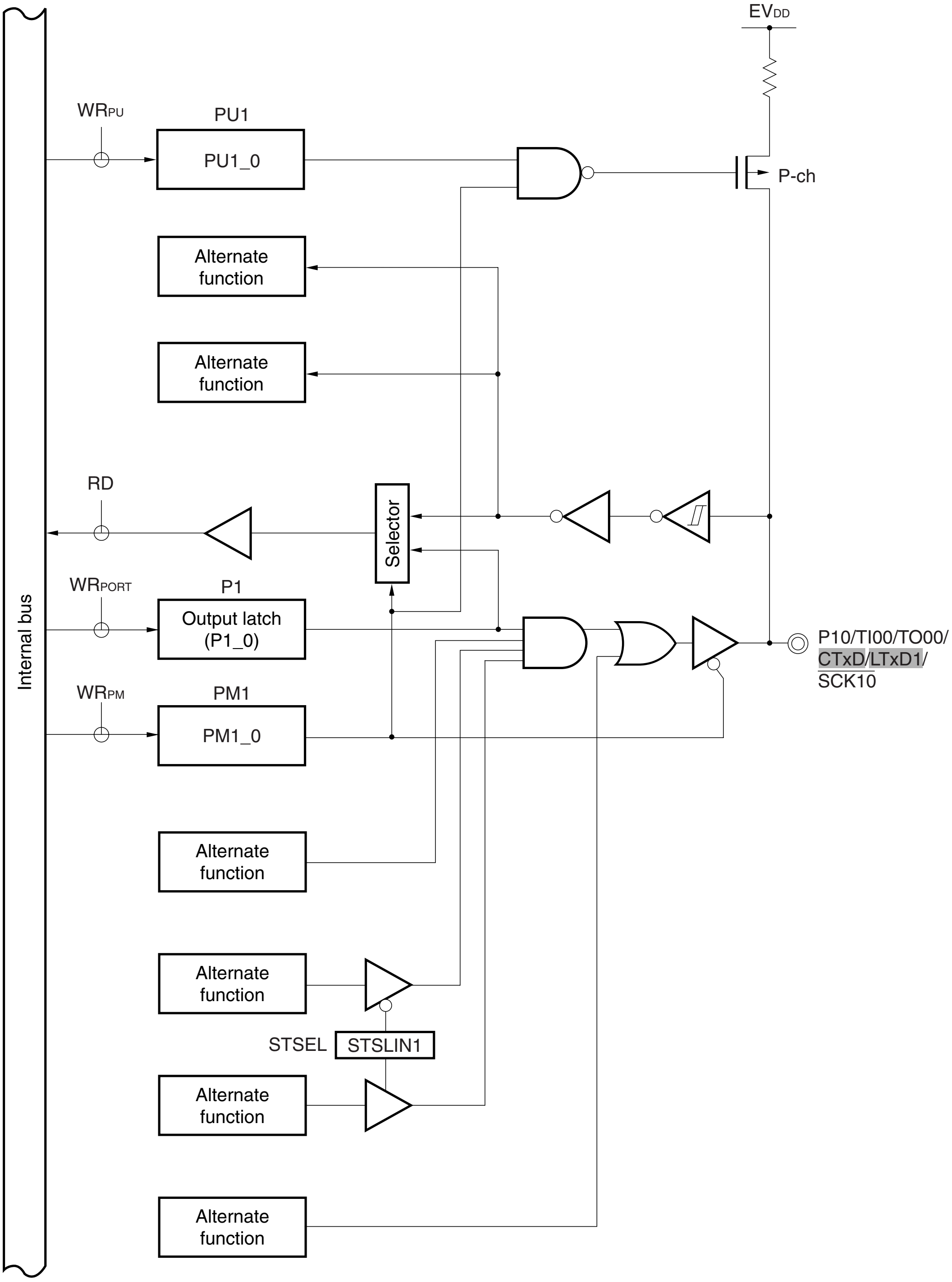
Figures 4-4 to 4-20 show block diagrams of port 1.

Cautions 1. To use P10/TI00/$\overline{\text{SCK10}}$/TO00/LTxD1/CTxD, P11/TI02/SI10/LRxD1/CRxD/INTPLR1/TO02, P12/INTP3/TI16/SO10/ TO16, P15/TI10/SO00/TO10, P16/TI12/SI00/TO12, or P17/TI14/$\overline{\text{SCK00}}$/TO14 as a general-purpose port, note the serial array unit setting. For details, see Table 11-8 Relationship Between Register Settings and Pins (Channel 0 of Unit 0: CSI00, STSCSI00 = 0) and Table 11-11 Relationship Between Register Settings and Pins (Channel 0 of Unit 1: CSI10).

2. To use P10/TI00/$\overline{\text{SCK10}}$/TO00/LTxD1/CTxD, P11/TI02/SI10/LRxD1/CRxD/INTPLR1/TO02, P12/INTP3/TI16/SO10/TO16, P13/TI04/LTxD0/TO04, P14/TI06/LRxD0/INTPLR0/TO06, P15/TI10/SO00/TO10, P16/TI12/SI00/TO12, or P17/TI14/$\overline{\text{SCK00}}$/TO14 as a general-purpose port, configure bits 0, 2, 4 and 6 (TO00, TO02, TO04, TO06) of timer output register 0 (TO0), bits 0, 2, 4, and 6 (TO10, TO12, TO14, TO16) of timer output register 1 (TO1), bits 0, 2, 4 and 6 (TOE0_0, TOE0_2, TOE0_4, TOE0_6) of timer output enable register 0 (TOE0), and bits 0, 2, 4, and 6 (TOE1_0, TOE1_2, TOE1_4, TOE1_6) of timer output enable register 1 (TOE1) to "0", which is the same as their default status setting.

3. The shaded pins are provided at two ports. Select either port by using the corresponding register.

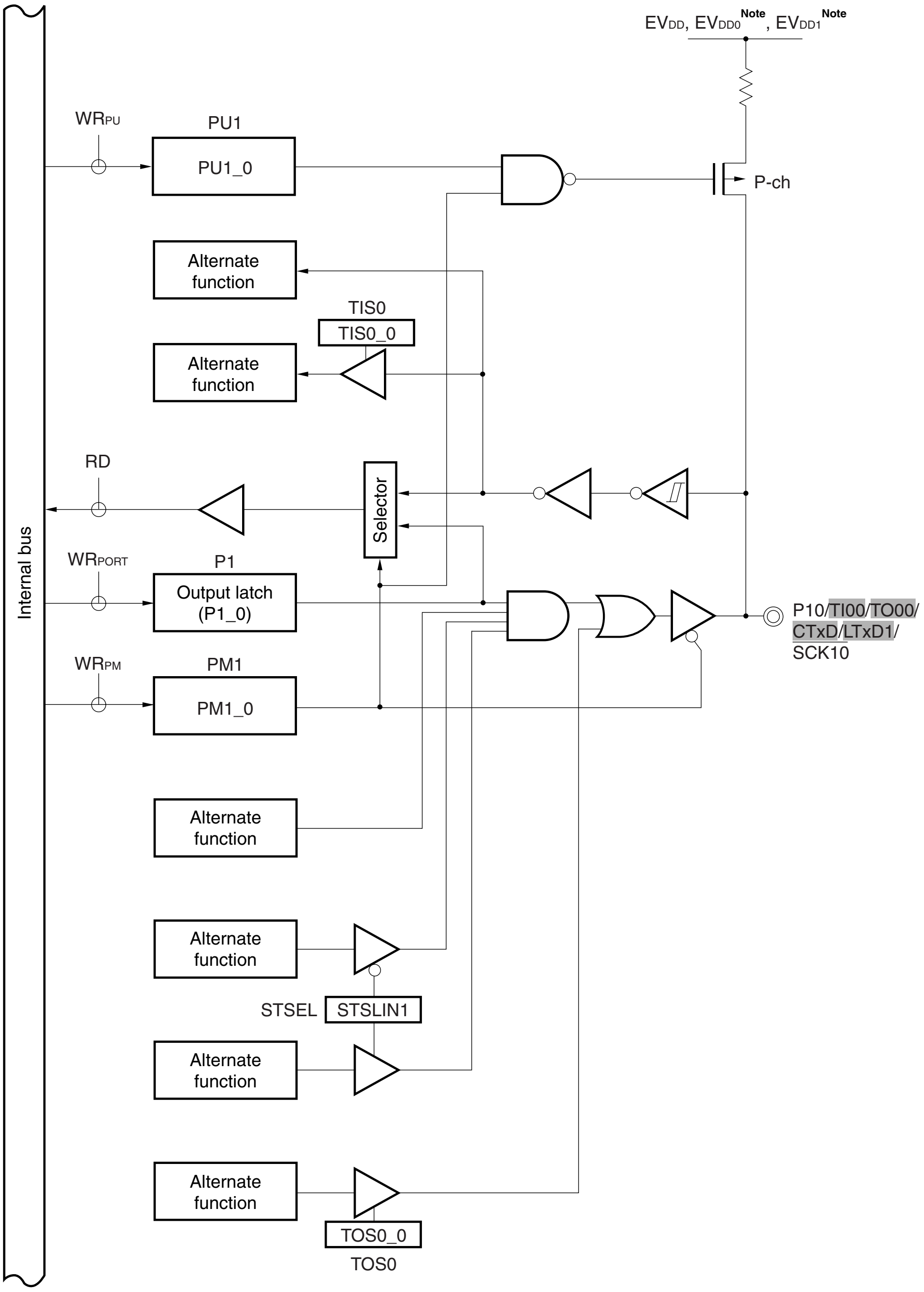
Figure 4-4. Block Diagram of P10 (78K0R/HC3, 78K0R/HE3)



Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
STSEL:	Serial communication pin select register
WR××:	Write signal

Figure 4-5. Block Diagram of P10 (78K0R/HF3, 78K0R/HG3)

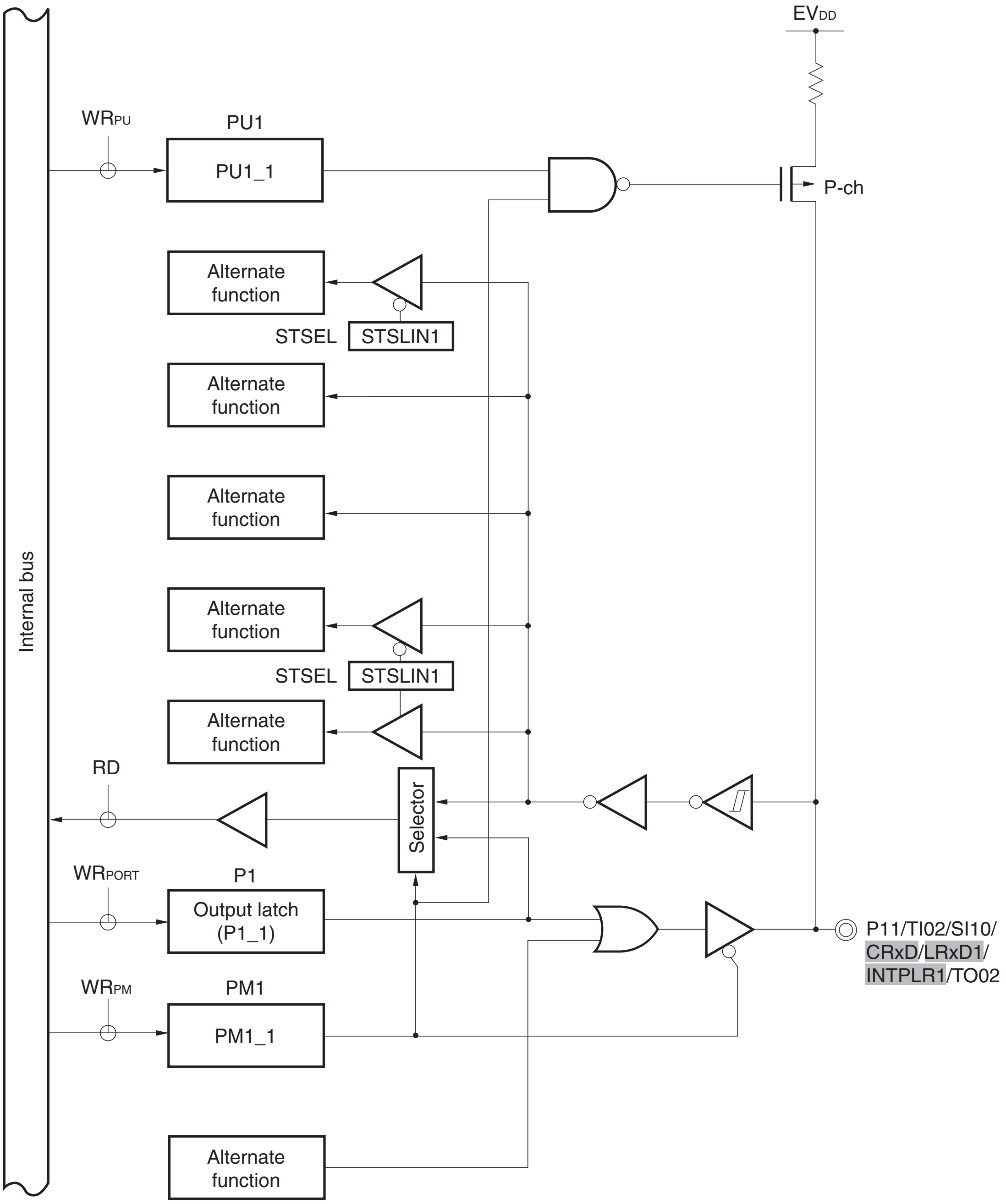


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
STSEL:	Serial communication pin select register
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

<R> **Figure 4-6. Block Diagram of P11 (78K0R/HC3, 78K0R/HE3)**

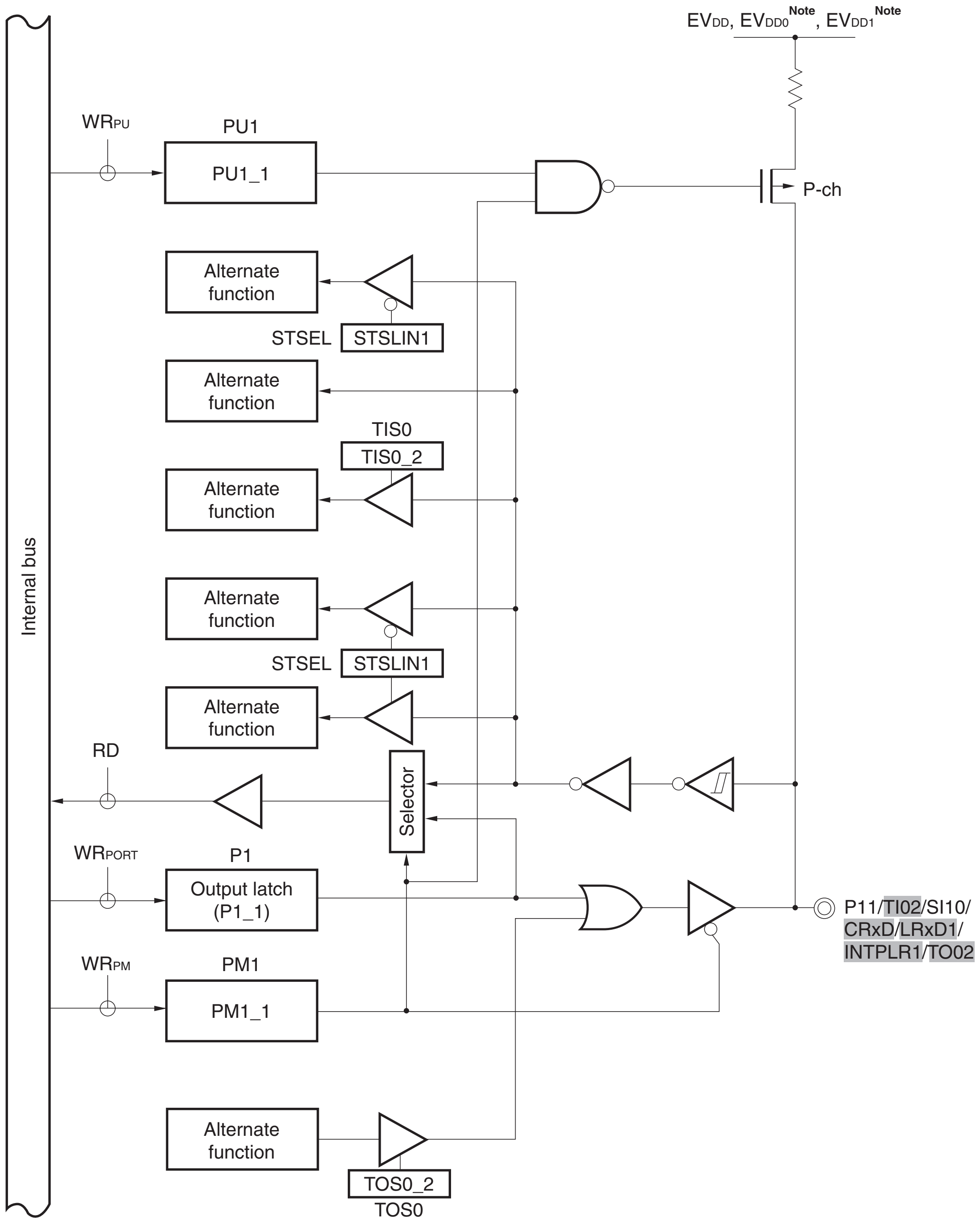


Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
STSEL:	Serial communication pin select register
WR××:	Write signal

<R>

Figure 4-7. Block Diagram of P11 (78K0R/HF3, 78K0R/HG3)

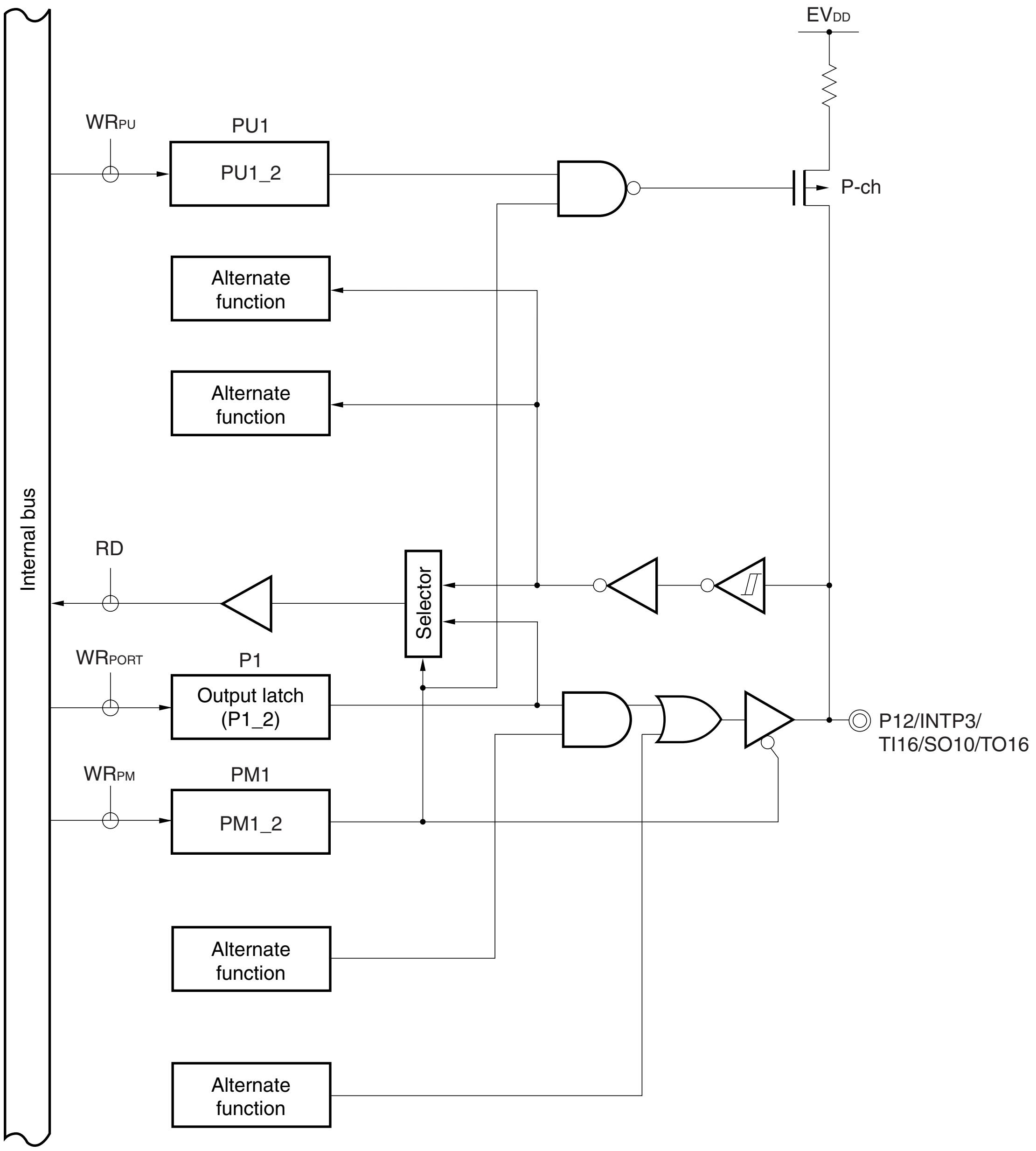


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

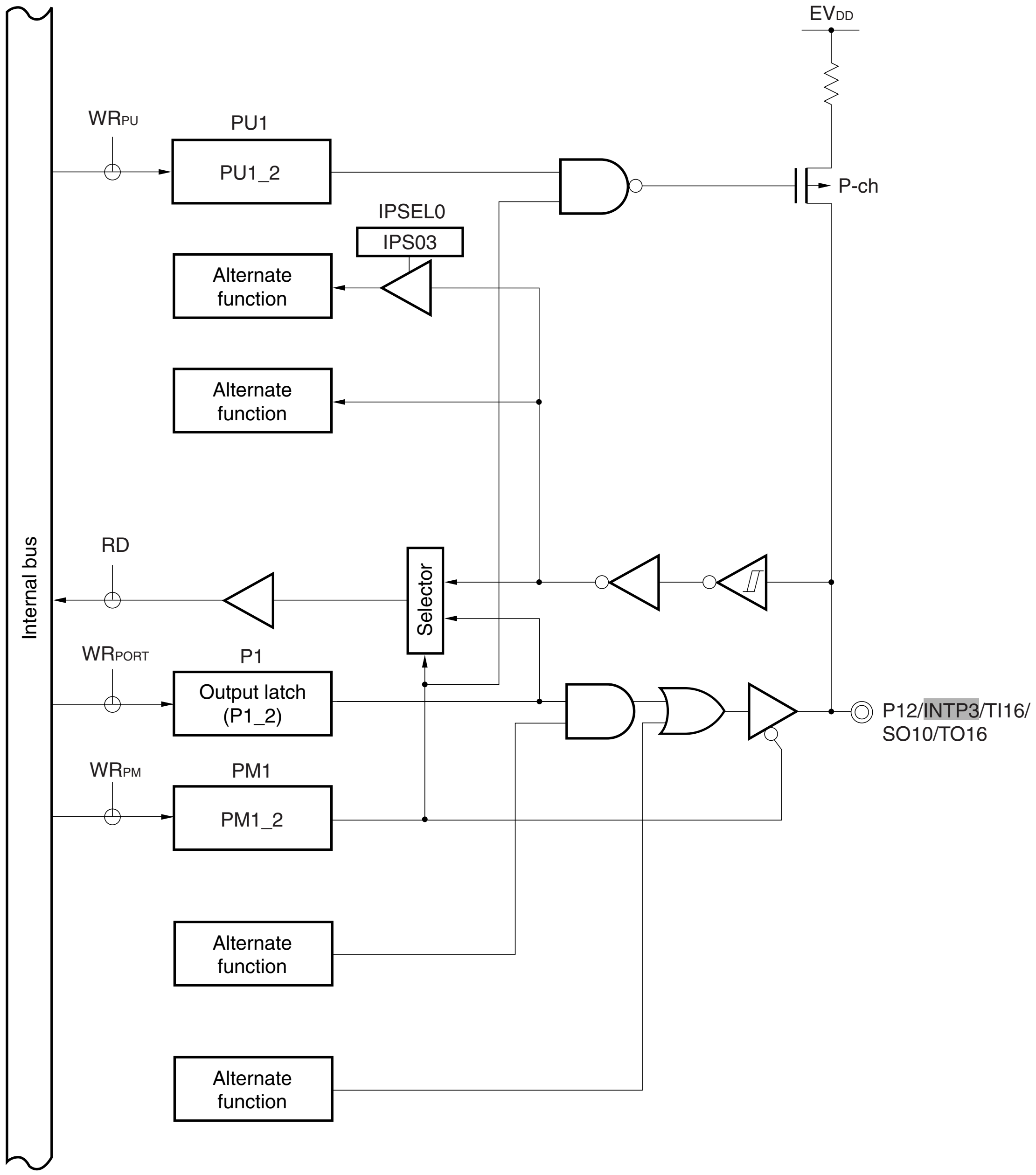
P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
STSEL:	Serial communication pin select register
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

Figure 4-8. Block Diagram of P12 (78K0R/HC3)



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

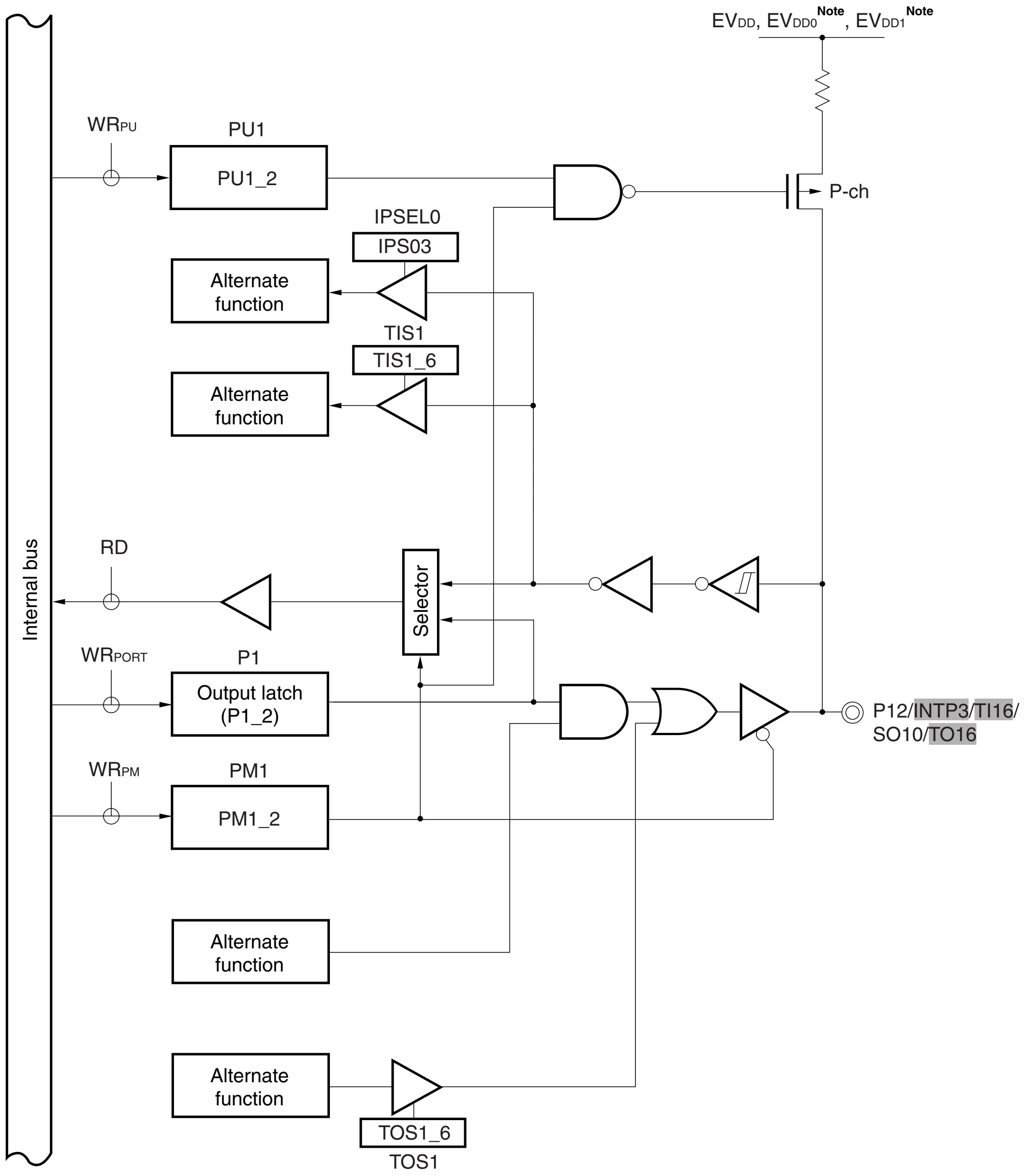
Figure 4-9. Block Diagram of P12 (78K0R/HE3)



Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

IPSEL0:	External interrupt input pin selection register 0
P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
WR××:	Write signal

Figure 4-10. Block Diagram of P12 (78K0R/HF3, 78K0R/HG3)

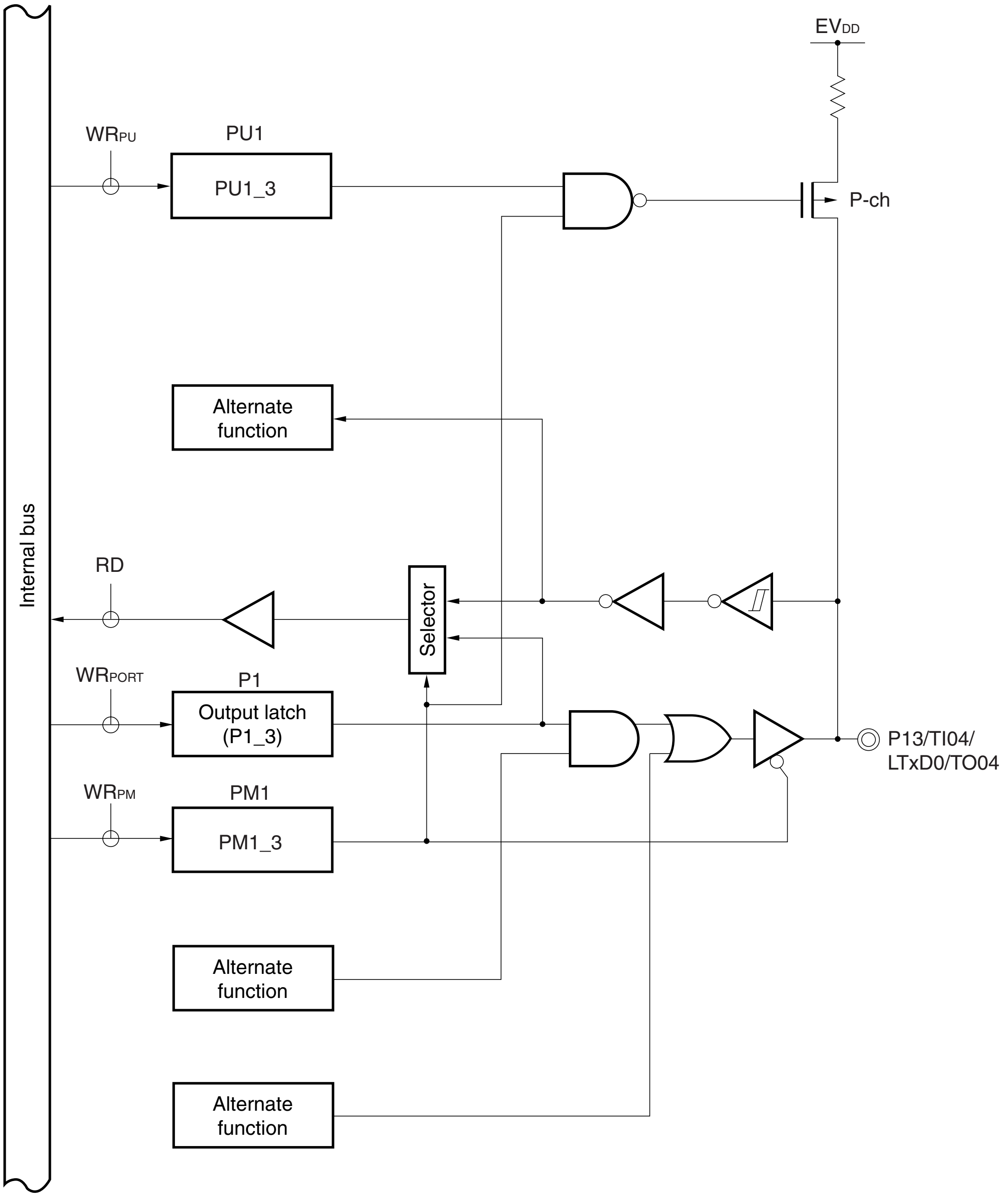


Note EVDD0, EVDD1 are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

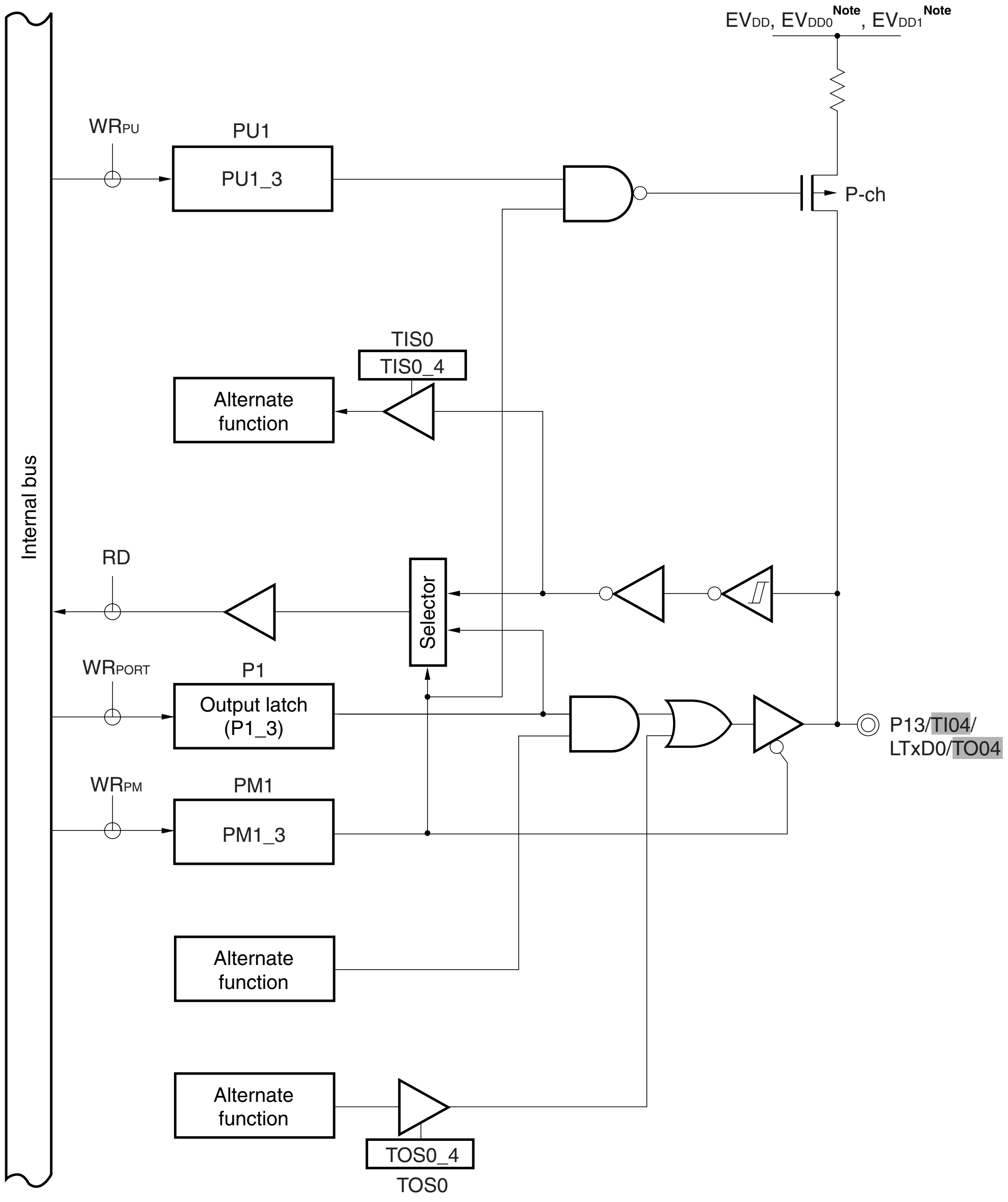
IPSEL0: External interrupt input pin selection register 0
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
TIS1: Timer input select register 1
TOS1: Timer output select register 1
WR××: Write signal

Figure 4-11. Block Diagram of P13 (78K0R/HC3, 78K0R/HE3)



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

Figure 4-12. Block Diagram of P13 (78K0R/HF3, 78K0R/HG3)

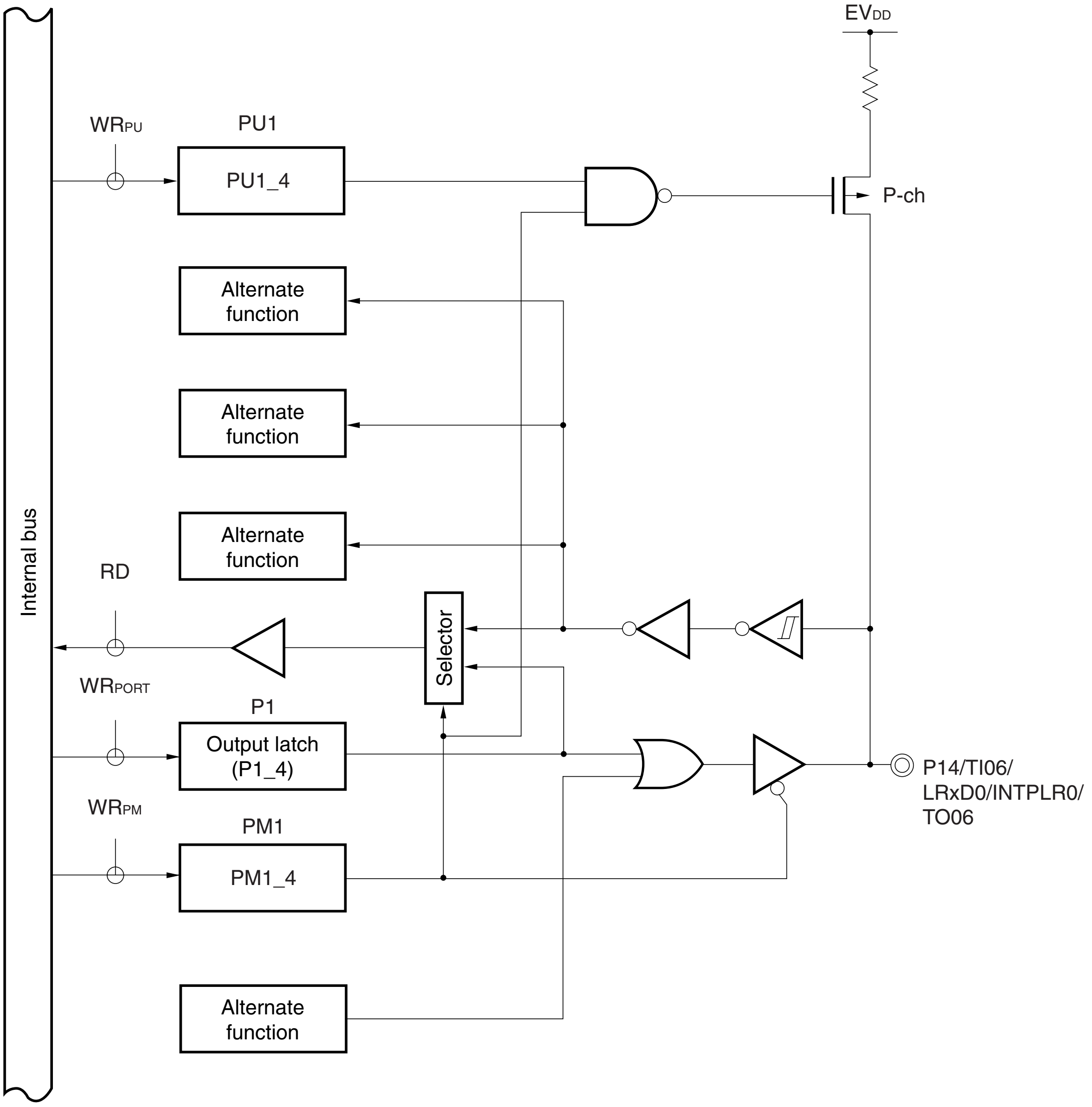


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

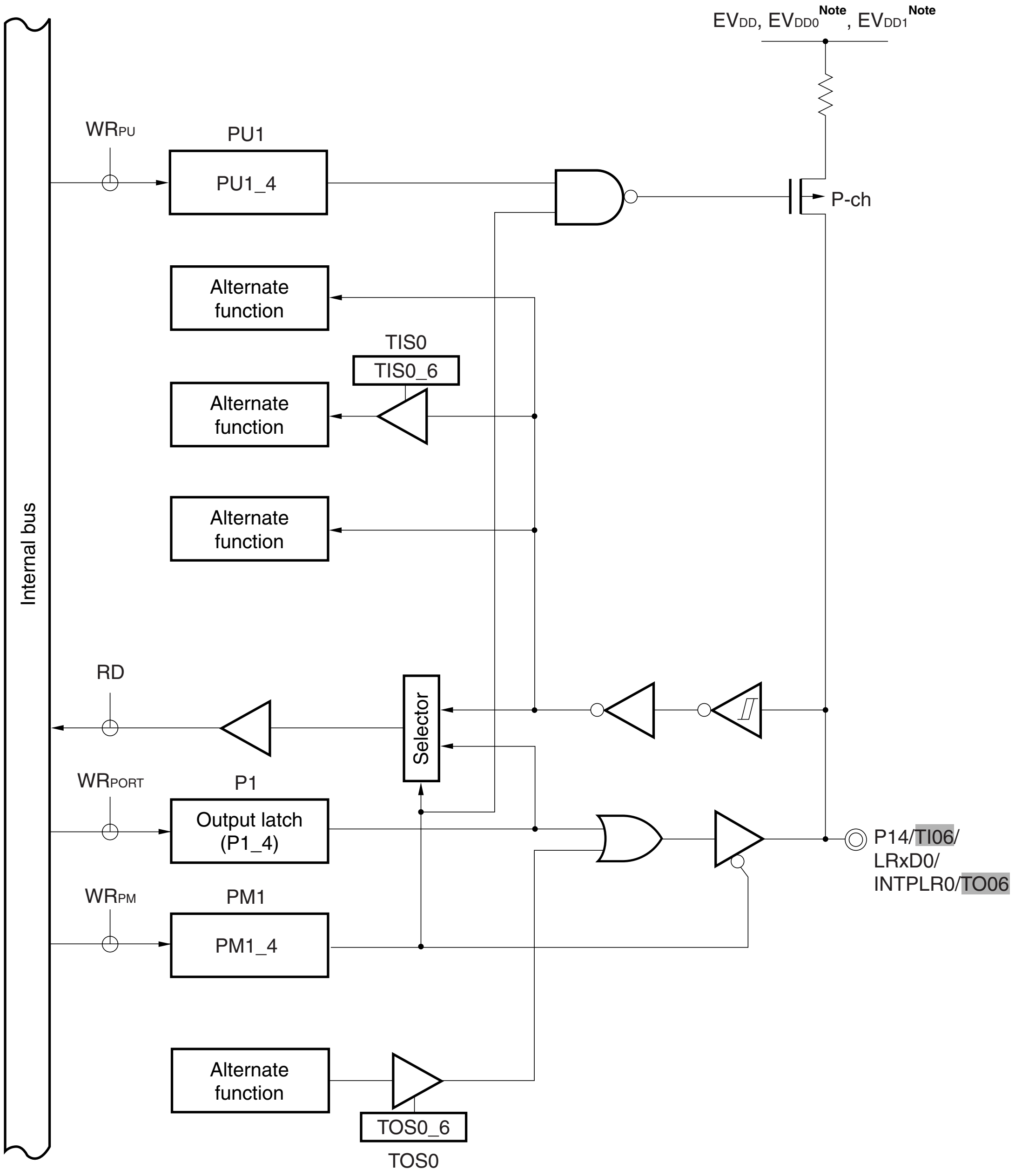
P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

Figure 4-13. Block Diagram of P14 (78K0R/HC3, 78K0R/HE3)



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

Figure 4-14. Block Diagram of P14 (78K0R/HF3, 78K0R/HG3)

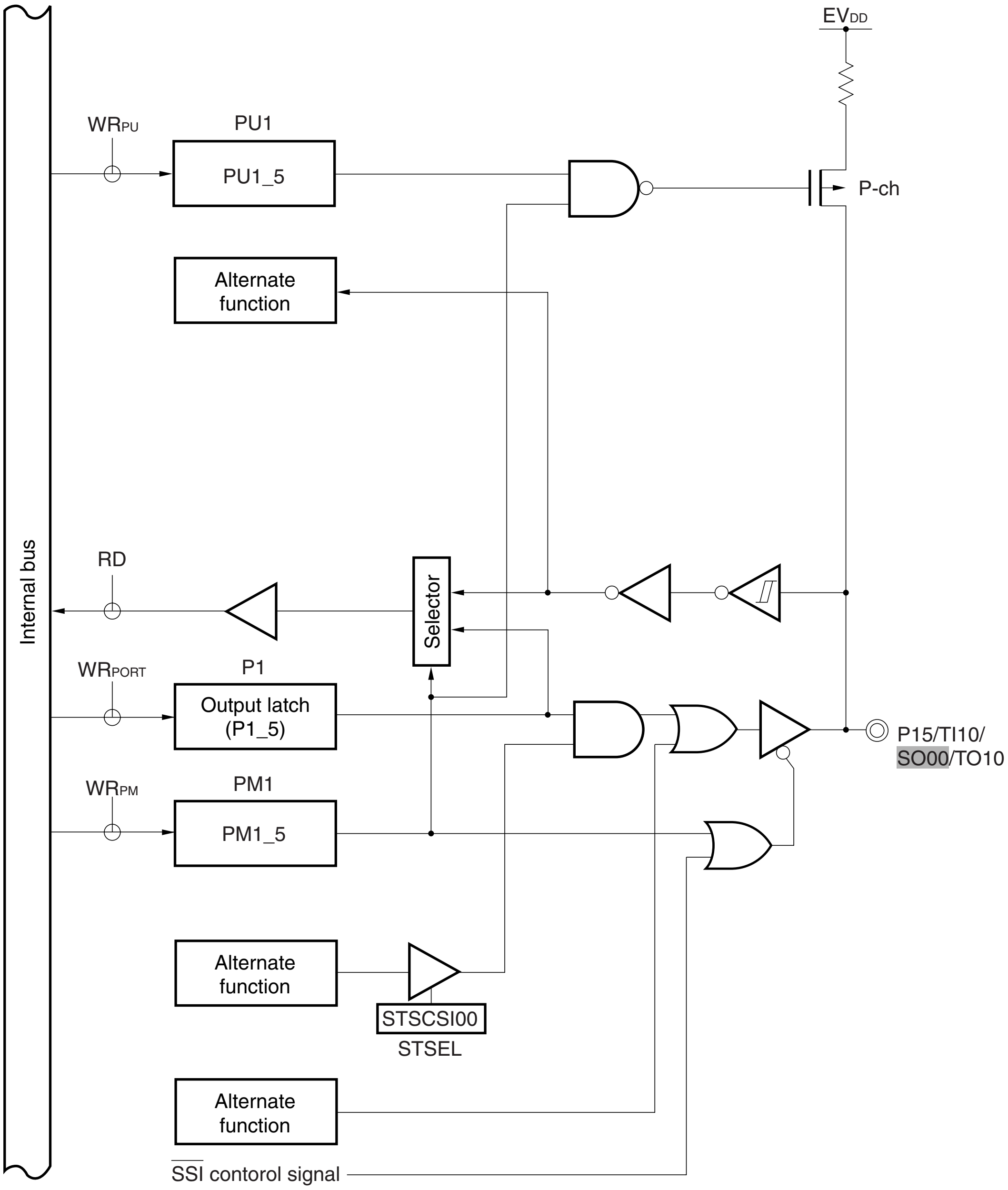


Note EVDD0, EVDD1 are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

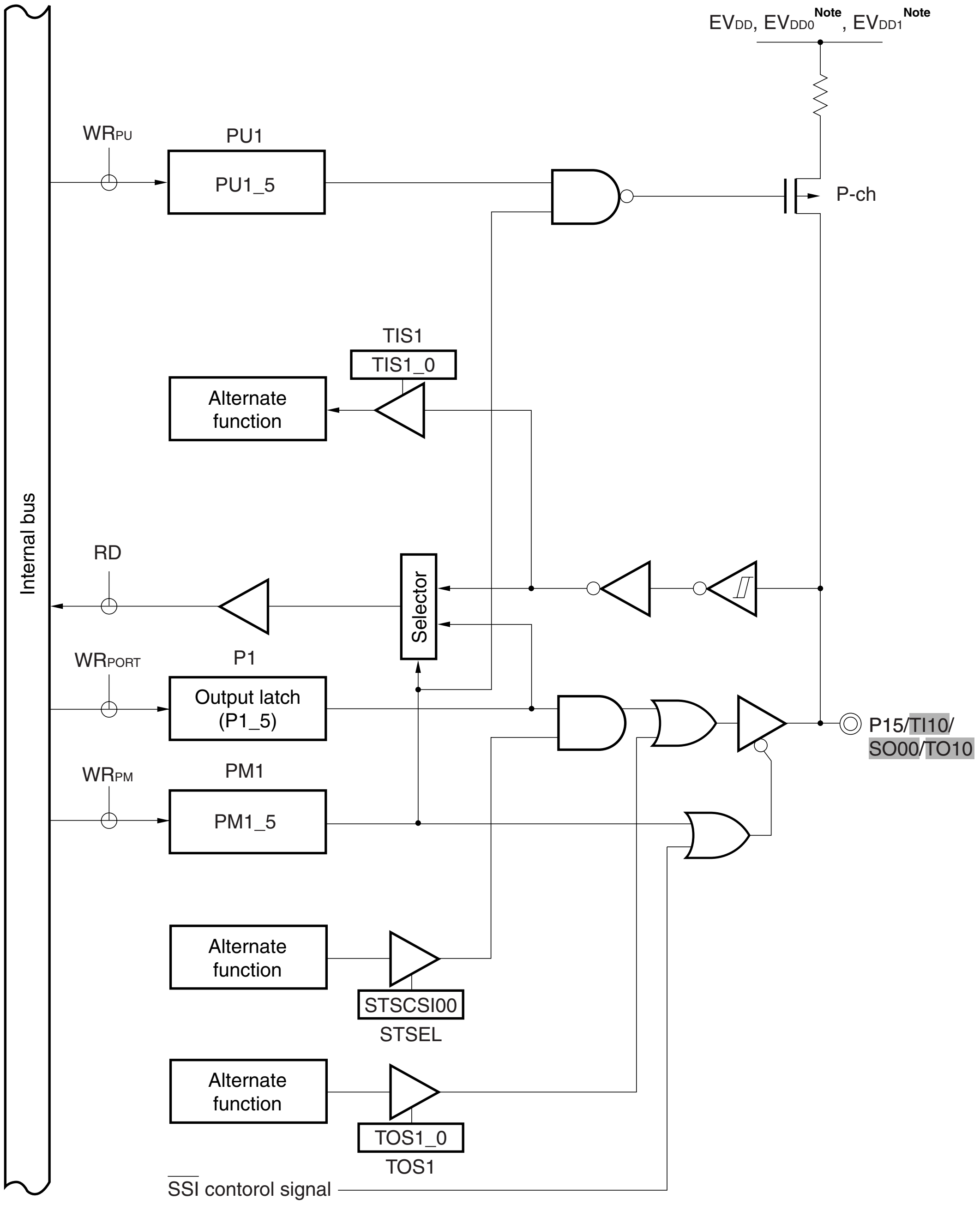
Figure 4-15. Block Diagram of P15 (78K0R/HC3, 78K0R/HE3)



Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
STSEL:	Serial communication pin select register
RD:	Read signal
WR××:	Write signal

Figure 4-16. Block Diagram of P15 (78K0R/HF3, 78K0R/HG3)

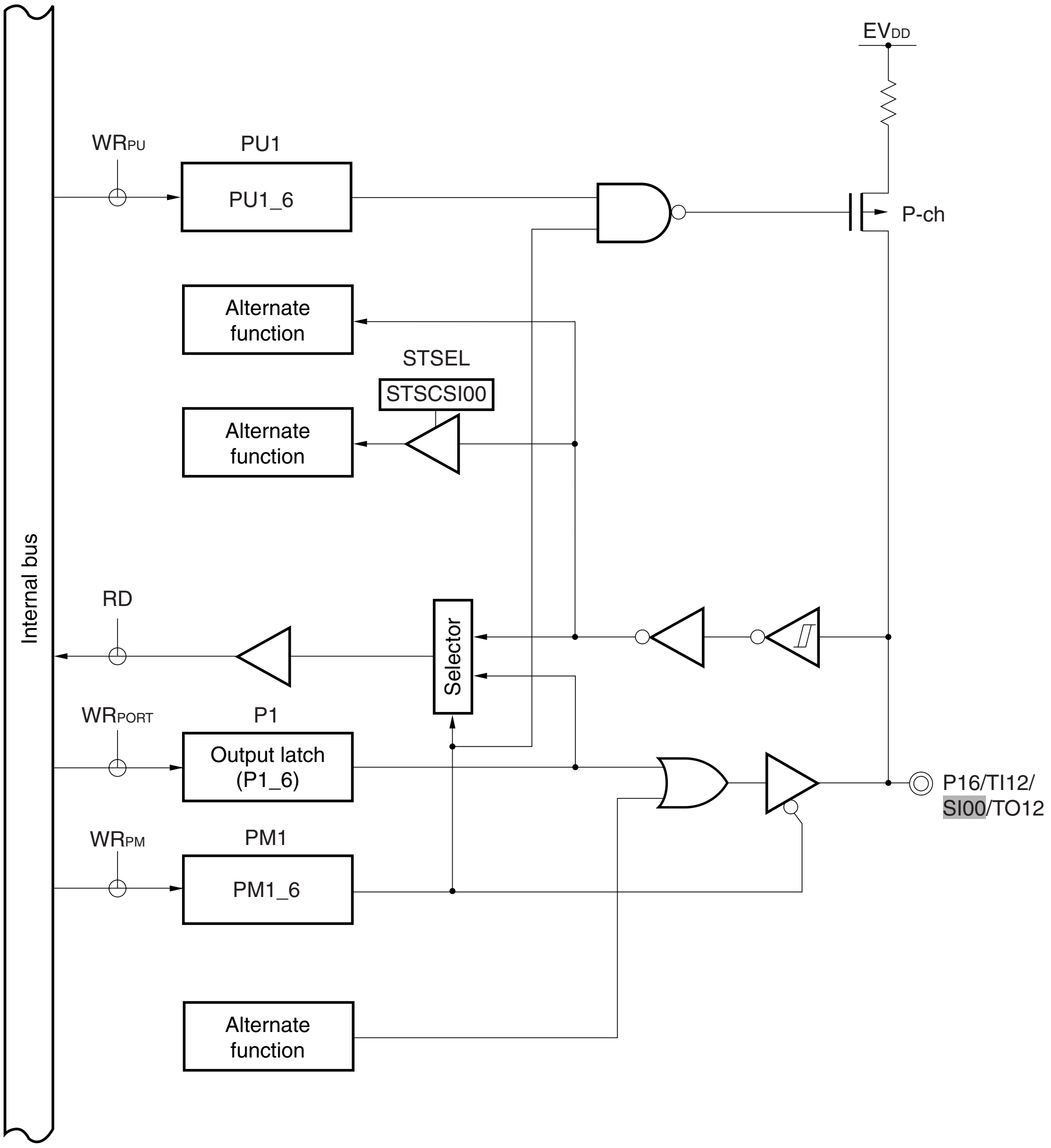


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
STSEL:	Serial communication pin select register
TIS1:	Timer input select register 1
TOS1:	Timer output select register 1
WR××:	Write signal

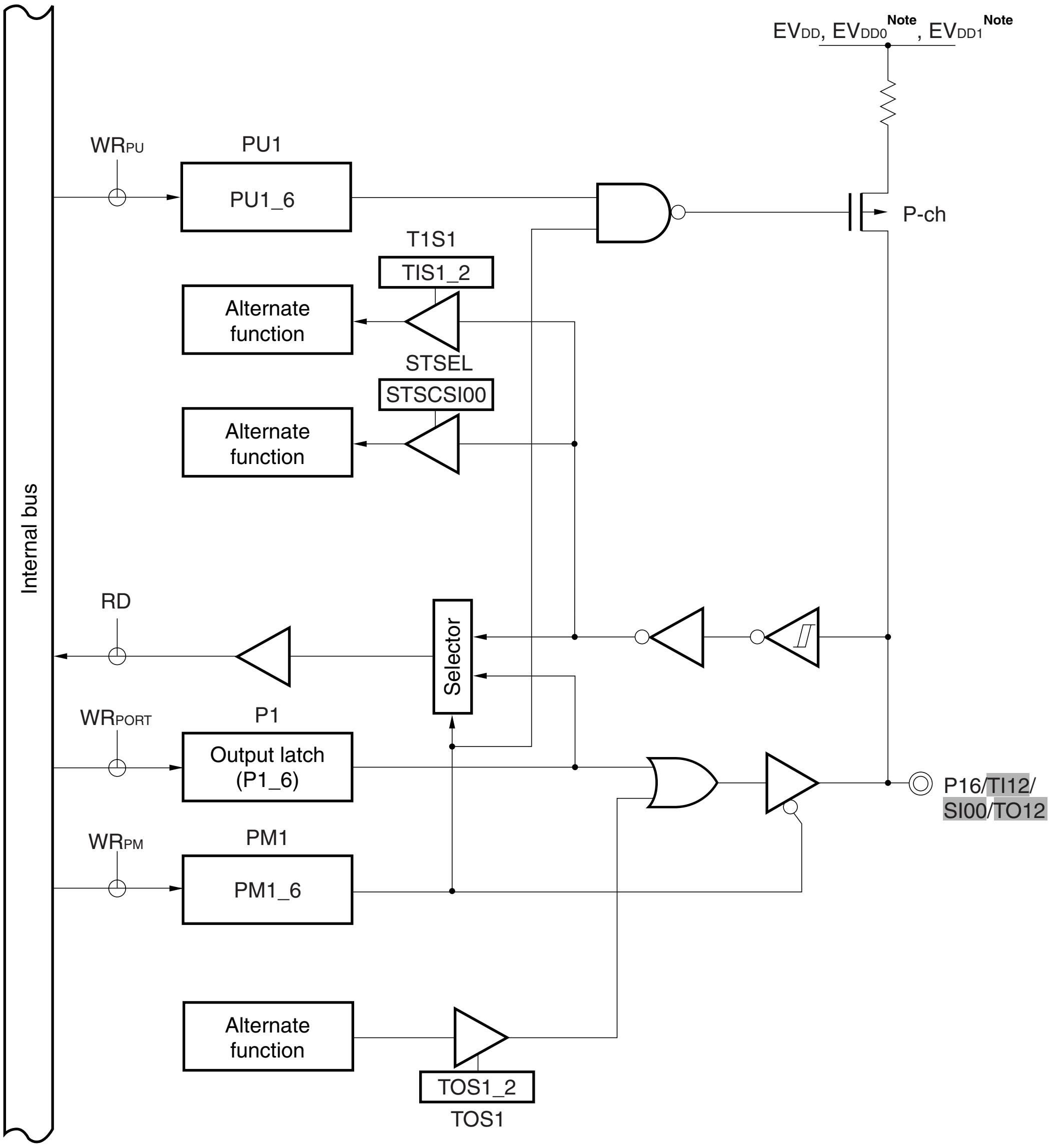
Figure 4-17. Block Diagram of P16 (78K0R/HC3, 78K0R/HE3)



Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
STSEL:	Serial communication pin select register
WR××:	Write signal

Figure 4-18. Block Diagram of P16 (78K0R/HF3, 78K0R/HG3)

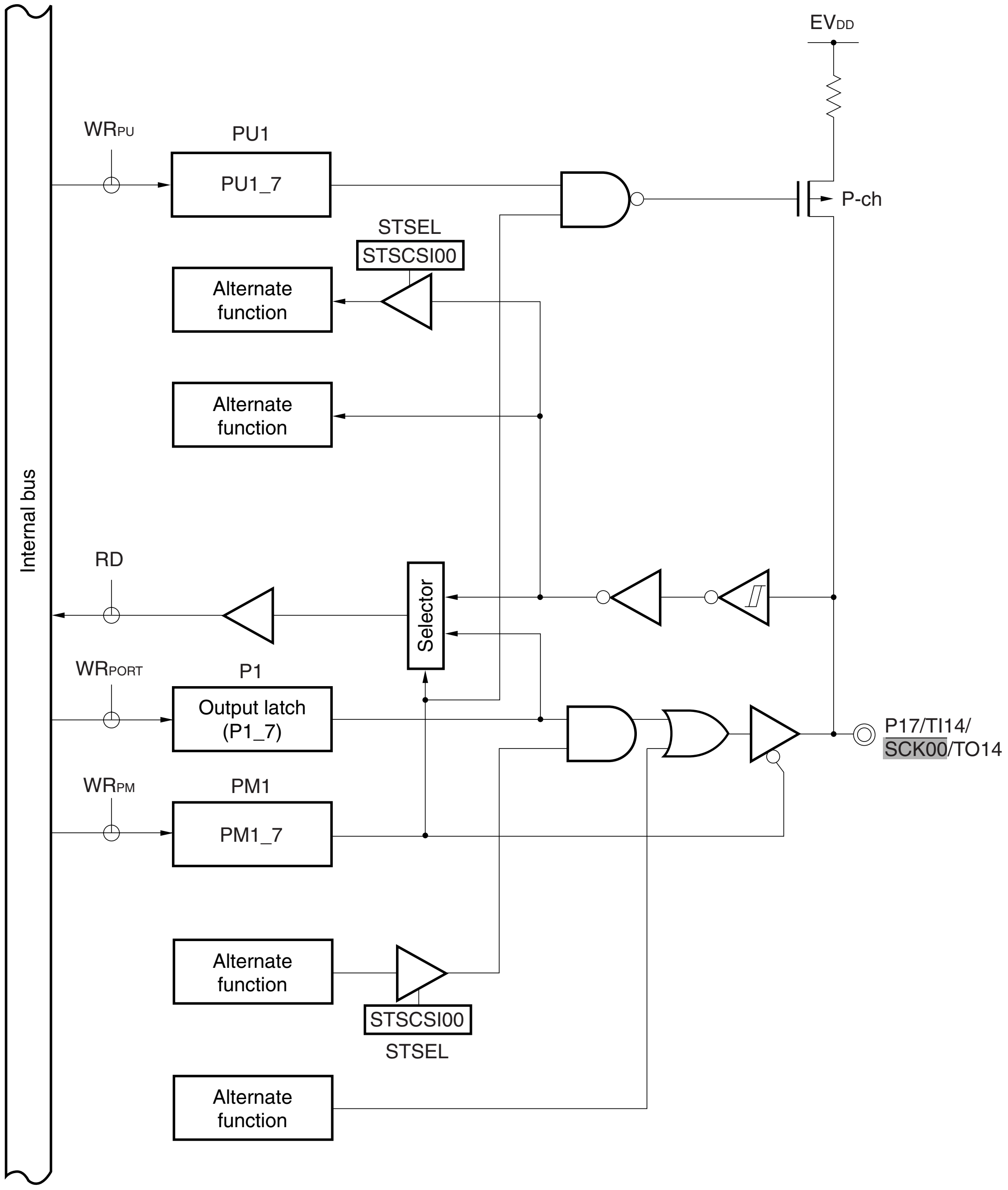


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
STSEL:	Serial communication pin select register
TIS1:	Timer input select register 1
TOS1:	Timer output select register 1
WR××:	Write signal

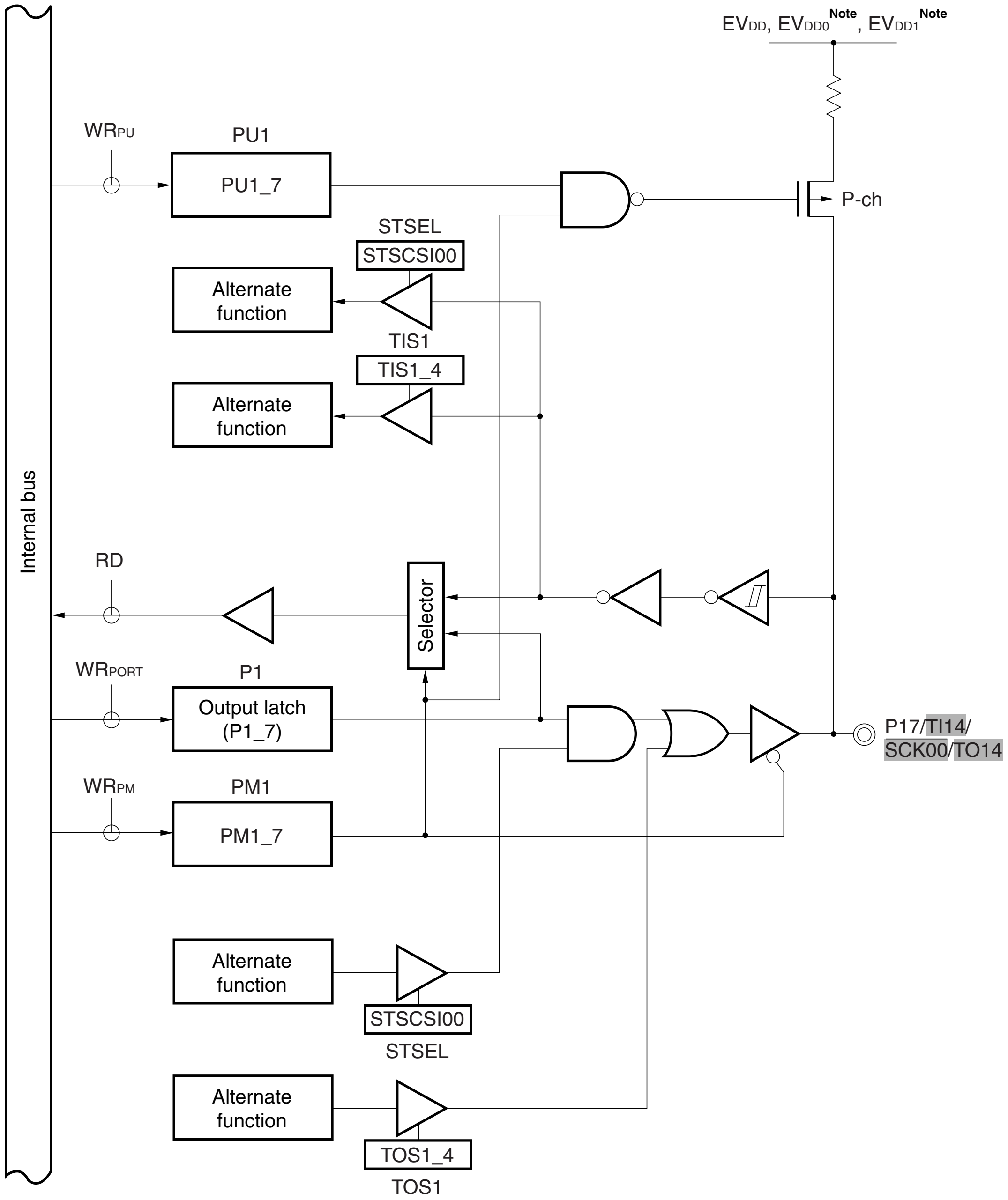
Figure 4-19. Block Diagram of P17 (78K0R/HC3, 78K0R/HE3)



Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
STSEL:	Serial communication pin select register
WR××:	Write signal

Figure 4-20. Block Diagram of P17 (78K0R/HF3, 78K0R/HG3)



Note EVDD0, EVDD1 are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
STSEL:	Serial communication pin select register
TIS1:	Timer input select register 1
TOS1:	Timer output select register 1
WR××:	Write signal

4.2.3 Port 3

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P30/INTP2/$\overline{\text{SSI00}}$/TI01/TO01	√	√	√	√
P31/INTP2/TI11/STOPST/TO11	√	√	√	√
P32/INTP4/TI13/TO13	√	√	√	√

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

Port 3 is an I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 to P32 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

This port can also be used for external interrupt request input, serial interface chip select input, timer I/O, and STOP status output.

Reset signal generation sets port 3 to input mode.

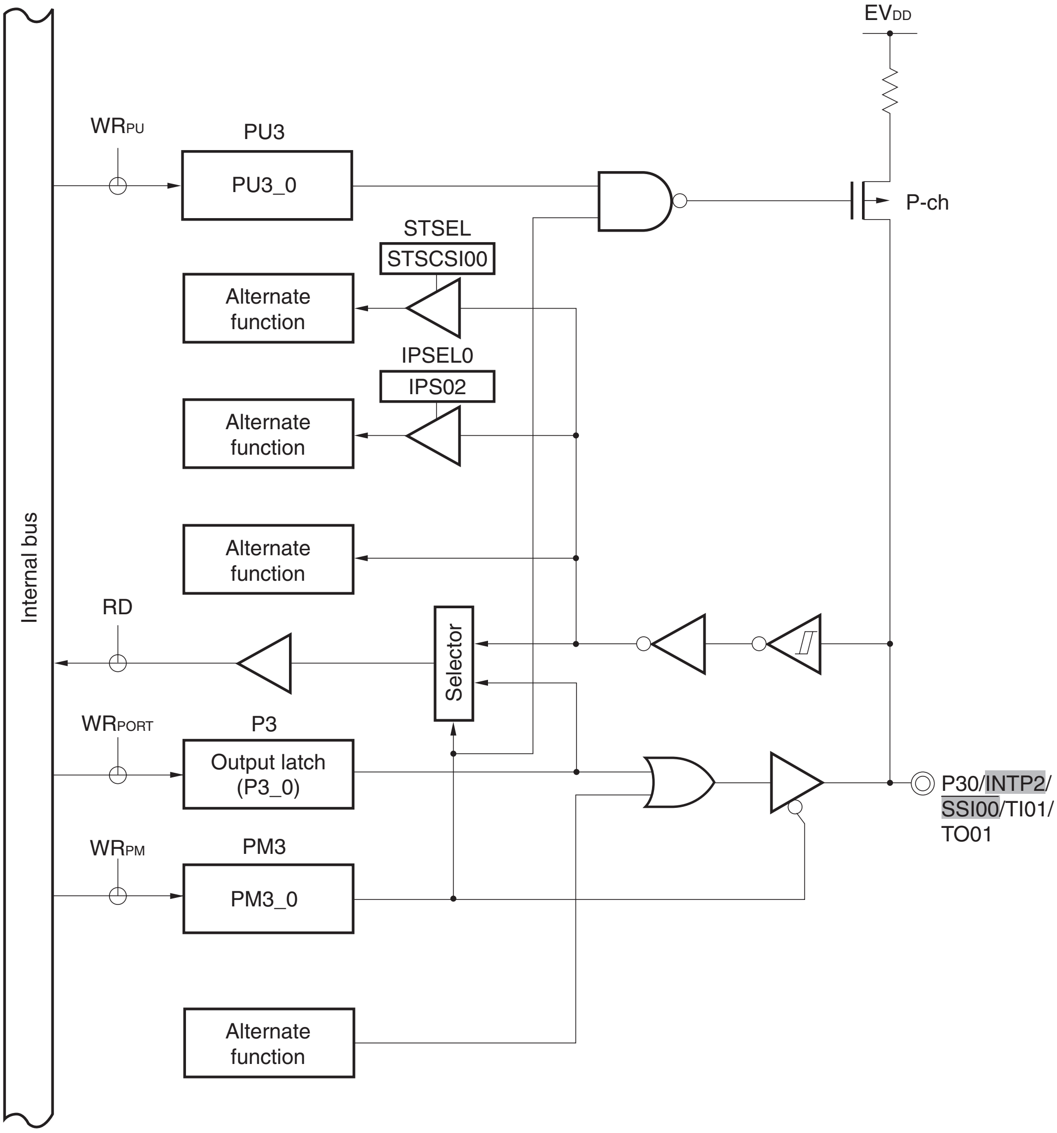
Figures 4-21 to 4-26 show block diagrams of port 3.

Cautions 1. To use P30/INTP2/$\overline{\text{SSI00}}$/TI01/TO01 as a general-purpose port, note the serial array unit setting. For details, see Table 11-8 Relationship Between Register Settings and Pins (Channel 0 of Unit 0: CSI00, STSCSI00 = 0).

2. To use P30/INTP2/$\overline{\text{SSI00}}$/TI01/TO01, P31/INTP2/TI11/STOPST/TO11, P32/INTP4/TI13/TO13 as a general-purpose port, configure bits 1 (TO01) of timer output register 0 (TO0) and bits 1 and 3 (TO11 and TO13) of timer output enable register 1 (TO1) and bit 1 (TOE0_1) of timer output enable register 0 (TOE0) and bits 1 and 3 (TOE1_1 and TOE1_3) of timer output enable register 1 (TOE1) to "0", which is the same as their default status setting.

3. The shaded pins are provided at two ports. Select either port by using the corresponding register.

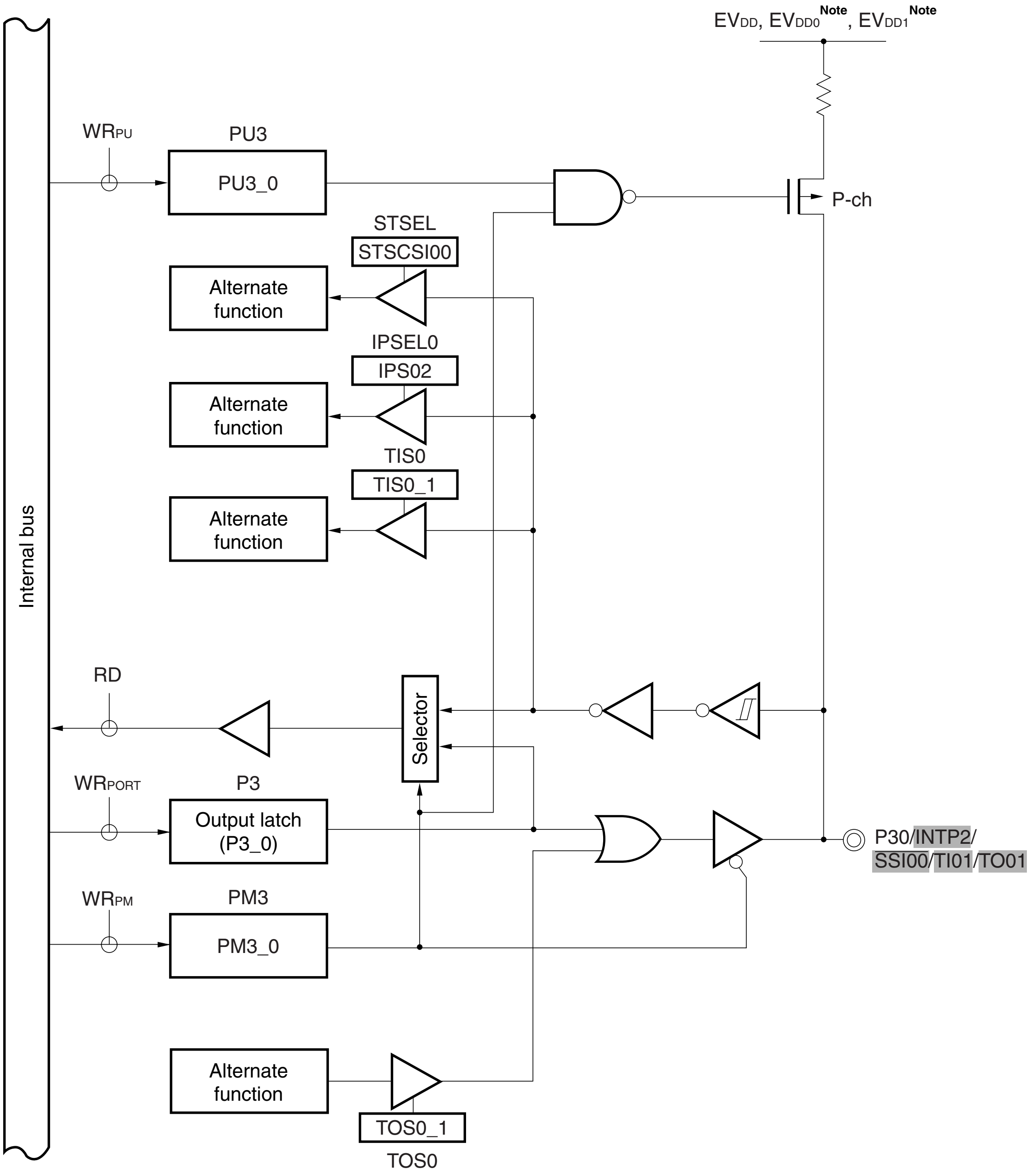
Figure 4-21. Block Diagram of P30 (78K0R/HC3, 78K0R/HE3)



Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

IPSEL0: External interrupt input pin selection register 0
P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
STSEL: Serial communication pin select register
WR××: Write signal

Figure 4-22. Block Diagram of P30 (78K0R/HF3, 78K0R/HG3)

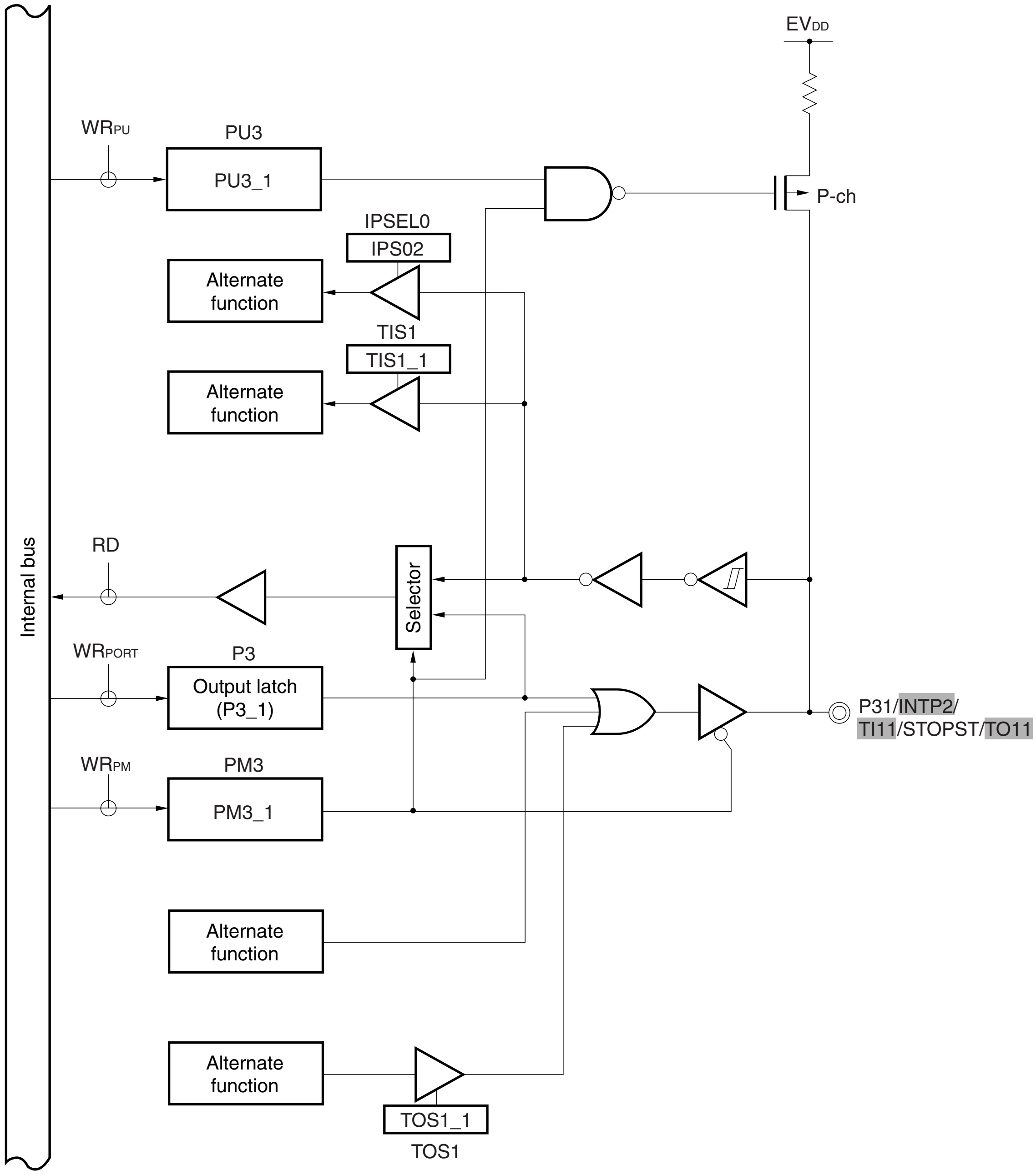


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

IPSEL0:	External interrupt input pin selection register 0
P3:	Port register 3
PU3:	Pull-up resistor option register 3
PM3:	Port mode register 3
RD:	Read signal
STSEL:	Serial communication pin select register
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

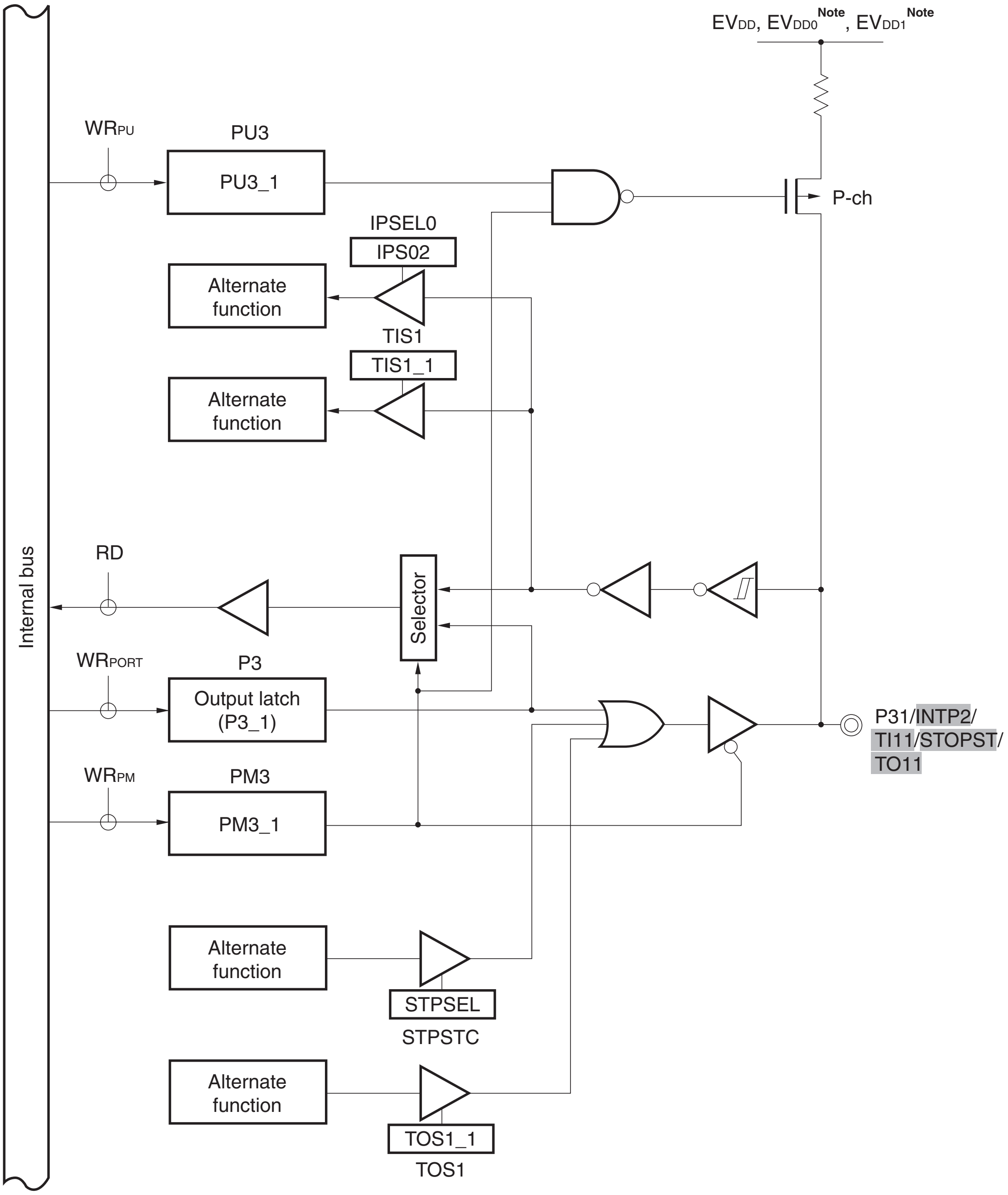
Figure 4-23. Block Diagram of P31 (78K0R/HC3)



Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

IPSEL0: External interrupt input pin selection register 0
P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
TIS1: Timer input select register 1
TOS1: Timer output select register 1
WR××: Write signal

<R> **Figure 4-24. Block Diagram of P31 (78K0R/HE3, 78K0R/HF3, 78K0R/HG3)**

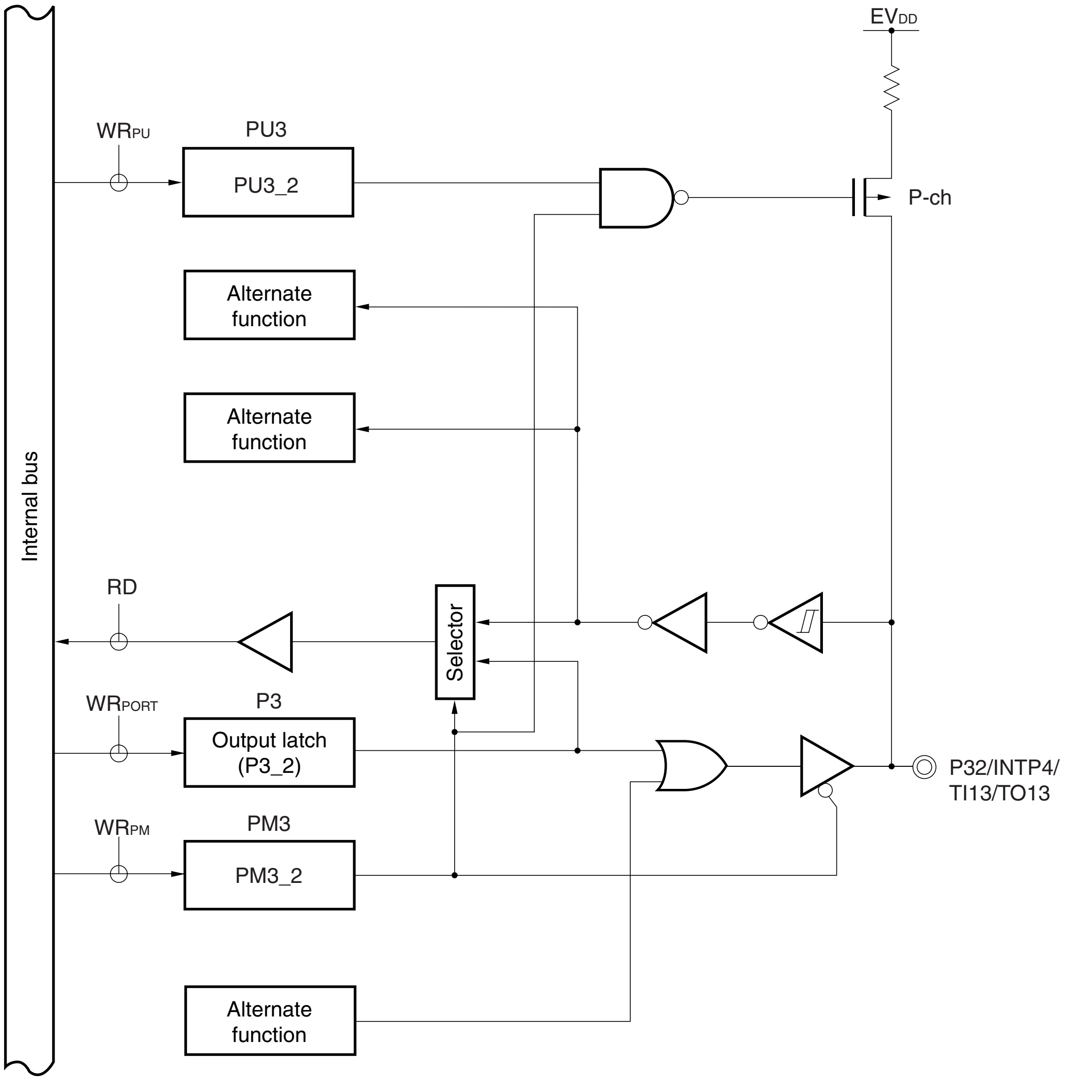


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

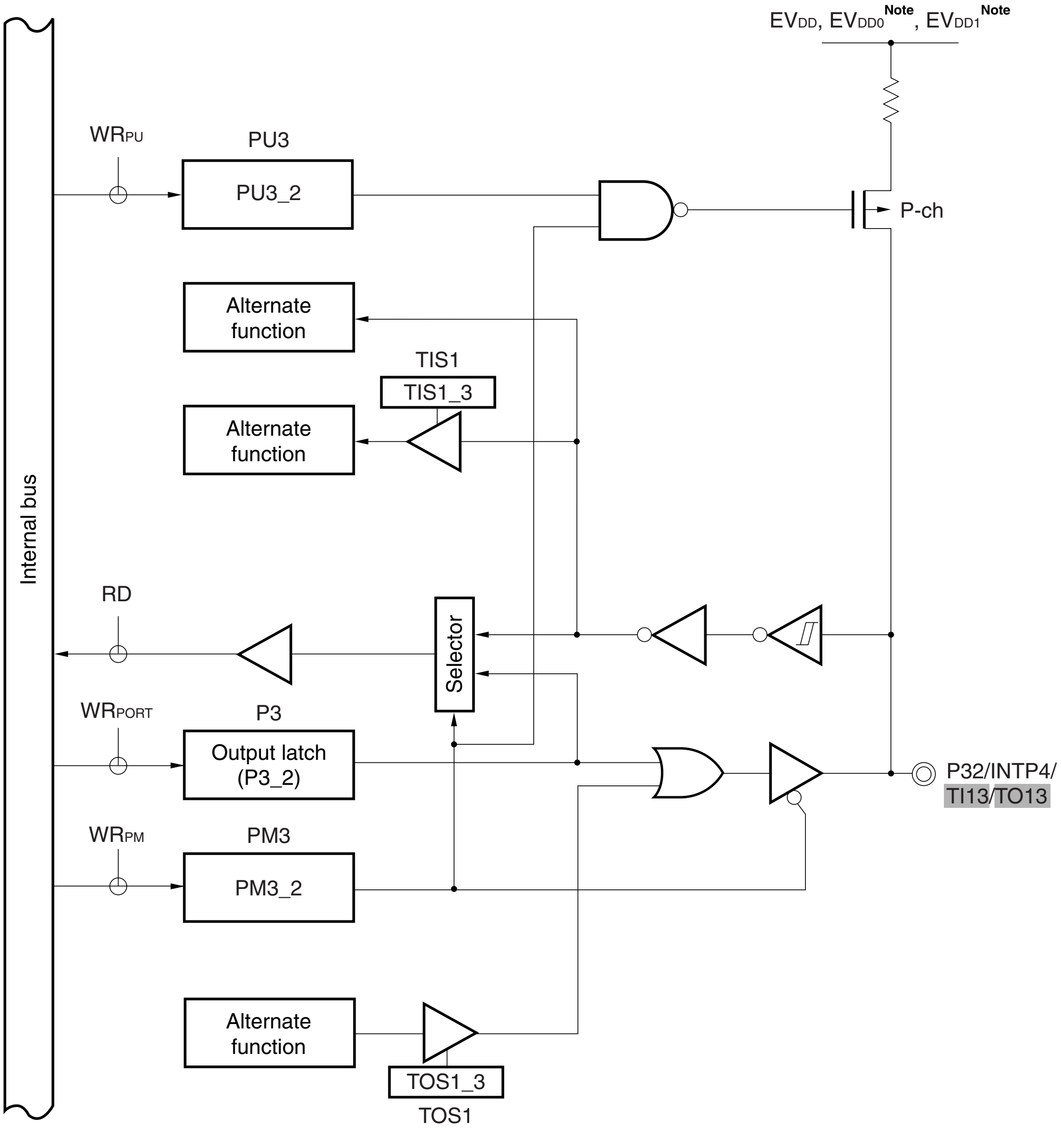
IPSEL0:	External interrupt input pin selection register 0
P3:	Port register 3
PU3:	Pull-up resistor option register 3
PM3:	Port mode register 3
RD:	Read signal
STPSTC:	STOP status output control register
TIS1:	Timer input select register 1
TOS1:	Timer output select register 1
WR××:	Write signal

Figure 4-25. Block Diagram of P32 (78K0R/HC3, 78K0R/HE3)



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

Figure 4-26. Block Diagram of P32 (78K0R/HF3, 78K0R/HG3)



Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P3:	Port register 3
PU3:	Pull-up resistor option register 3
PM3:	Port mode register 3
RD:	Read signal
TIS1:	Timer input select register 1
TOS1:	Timer output select register 1
WR××:	Write signal

4.2.4 Port 4

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P40/TOOL0/TI05/TO05	√	√	√	√
P41/TOOL1/TI07/TO07	√	√	√	√
P42/TxD2/SCL20	–	√	√	√
P43/RxD2/SDA20/INTPR2	–	√	√	√
P44/TI07/TO07	–	–	√	√
P45/TI10/TO10	–	–	√	√
P46/TI12/TO12	–	–	√	√
P47/INTP8	–	–	√	√

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 to P47 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4)[Note].

Output from the P42 and P43 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 4 (POM4).

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

Reset signal generation sets port 4 to input mode.

Figures 4-27 to 4-34 show block diagrams of port 4.

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

(Caution is given on the next page.)

Cautions 1. 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).

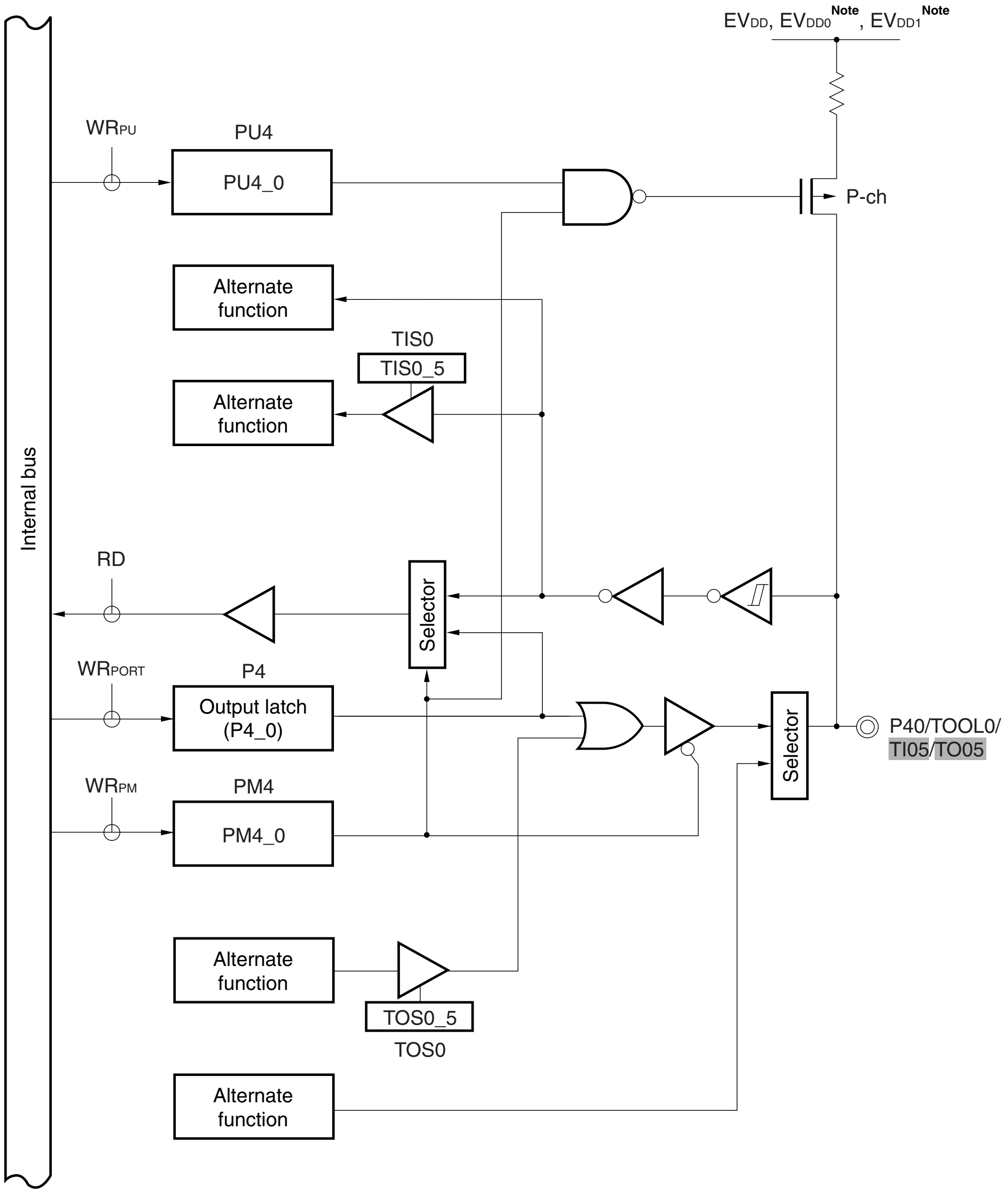
2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).

2. To use P42/TxD2/SCL20, or P43/RxD2/SDA20/INTPR2 as a general-purpose port, note the serial array unit 2 setting. For details, see Table 11-13 Relationship Between Register Settings and Pins (Channel 0 of Unit 2: UART2 Transmission, IIC20) and Table 11-14 Relationship Between Register Settings and Pins (Channel 1 of Unit 2: UART2 Reception).

3. To use P40/TOOL0/TI05/TO05, P41/TOOL1/TI07/TO07, P44/TI07/TO07, P45/TI10/TO10 or P46/TI12/TO12 as a general-purpose port, configure bits 5 and 7 (TO05, TO07) of timer output register 0 (TO0) and bits 0 and 2 (TO10, TO12) of timer output register 1 (TO1) and bits 5 and 7 (TOE0_5, TOE0_7) of timer output enable register 0 (TOE0) and bits 0 and 2 (TOE1_0, TOE1_2) of timer output enable register 1 (TOE1) to "0", which is the same as their default status setting.

4. The shaded pins are provided at two ports. Select either port by using the corresponding register.

Figure 4-27. Block Diagram of P40 (78K0R/HC3, 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)

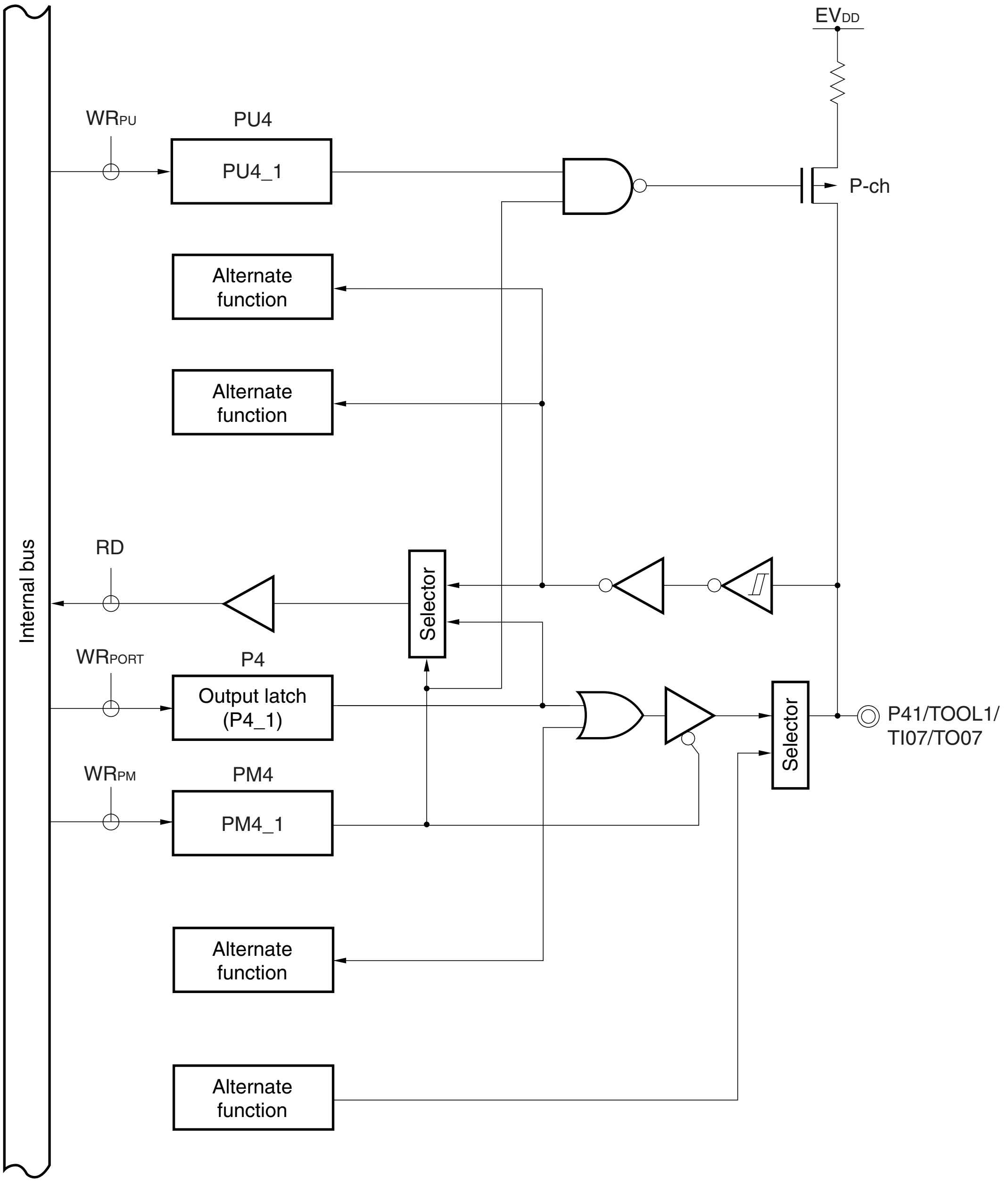


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

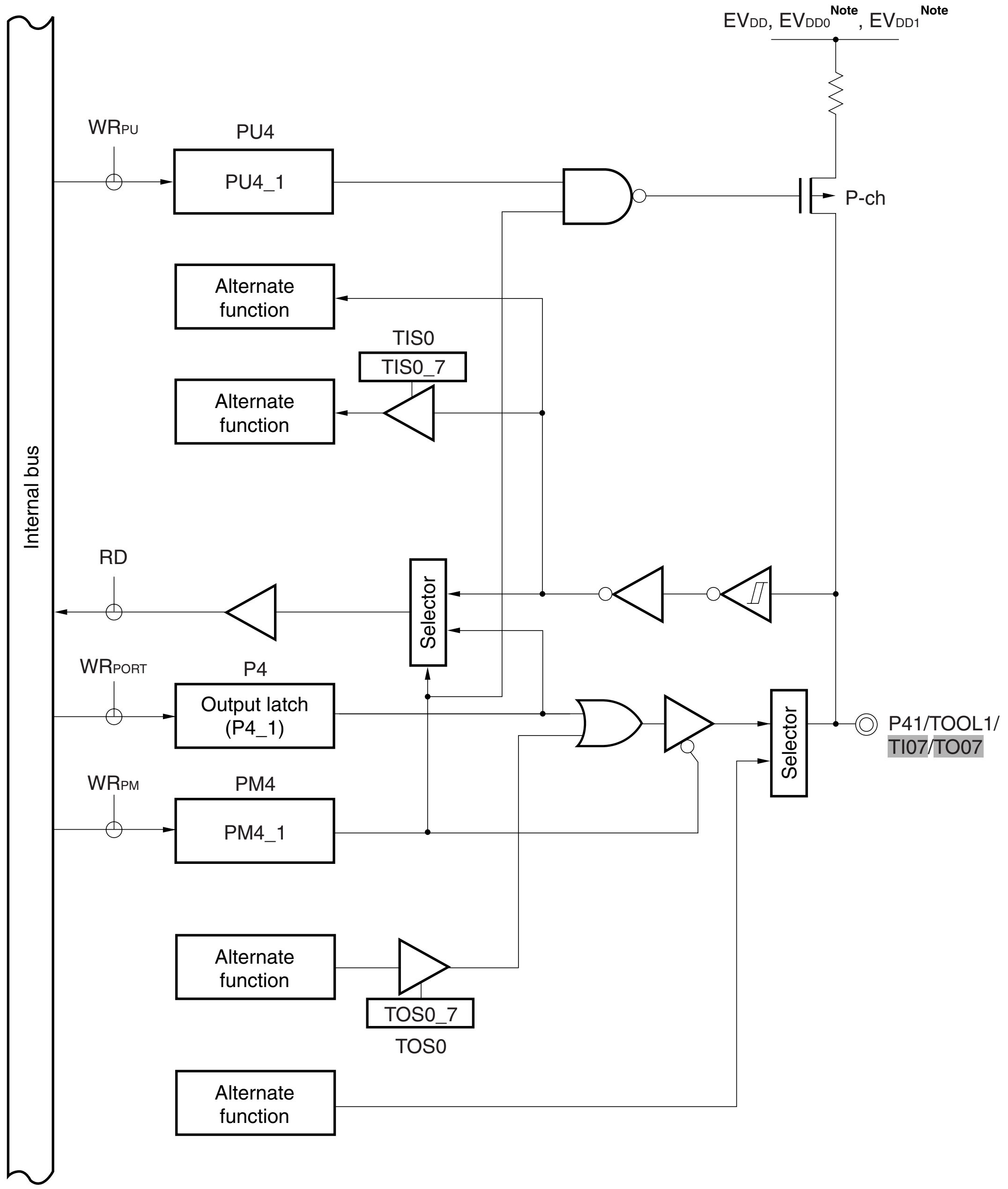
P4:	Port register 4
PU4:	Pull-up resistor option register 4
PM4:	Port mode register 4
RD:	Read signal
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

Figure 4-28. Block Diagram of P41 (78K0R/HC3, 78K0R/HE3)



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

Figure 4-29. Block Diagram of P41 (78K0R/HF3, 78K0R/HG3)

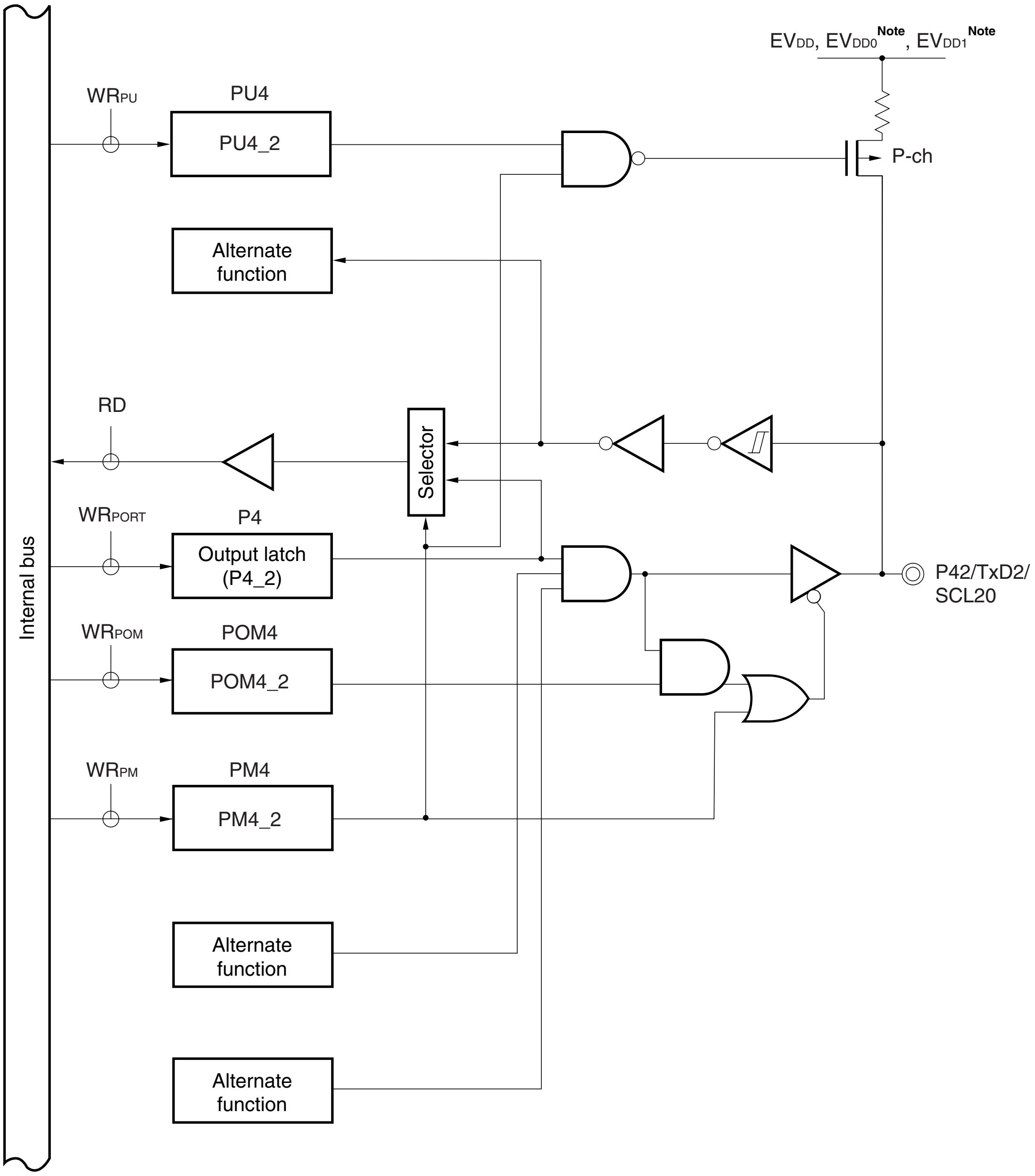


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P4:	Port register 4
PU4:	Pull-up resistor option register 4
PM4:	Port mode register 4
RD:	Read signal
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

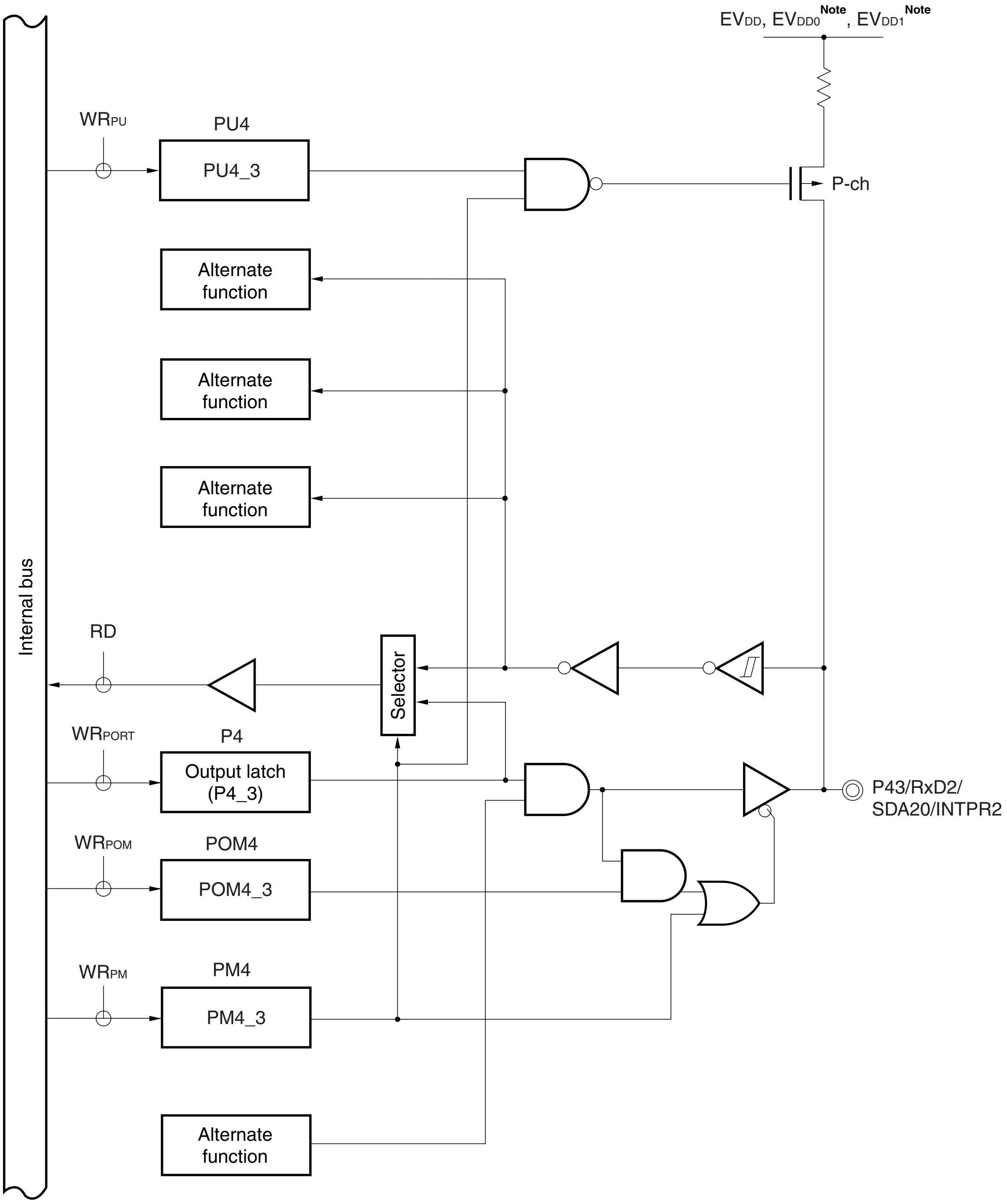
Figure 4-30. Block Diagram of P42 (78K0R/HE3, 78K0R/HF3, 78K0R/HG3)



Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

P4:	Port register 4
PU4:	Pull-up resistor option register 4
PM4:	Port mode register 4
POM4:	Port output mode register 4
RD:	Read signal
WR××:	Write signal

Figure 4-31. Block Diagram of P43 (78K0R/HE3, 78K0R/HF3, 78K0R/HG3)

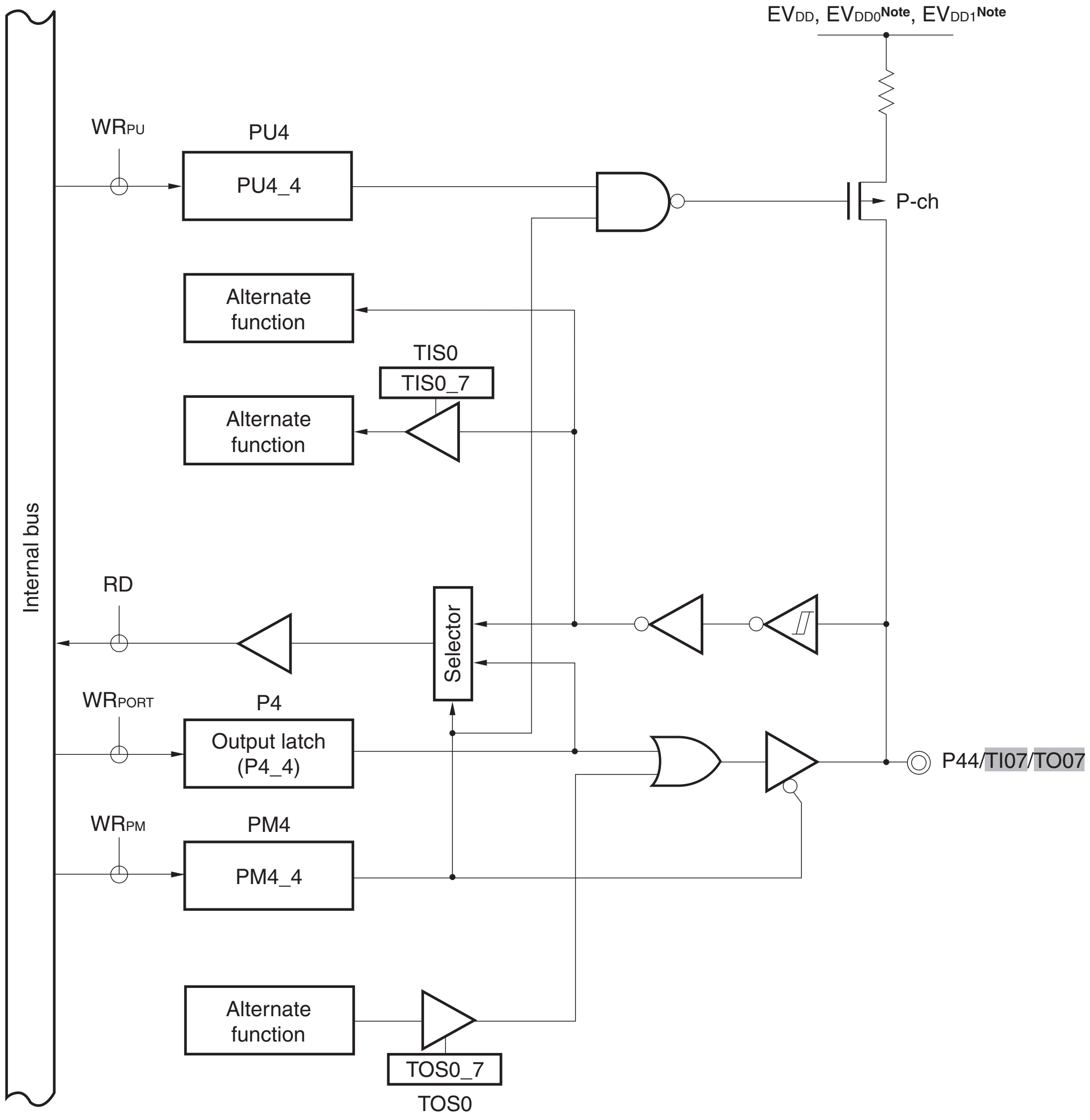


Note EVDD0, EVDD1 are 78K0R/HG3 only

P4:	Port register 4
PU4:	Pull-up resistor option register 4
PM4:	Port mode register 4
POM4:	Port output mode register 4
RD:	Read signal
WR××:	Write signal

<R>

Figure 4-32. Block Diagram of P44

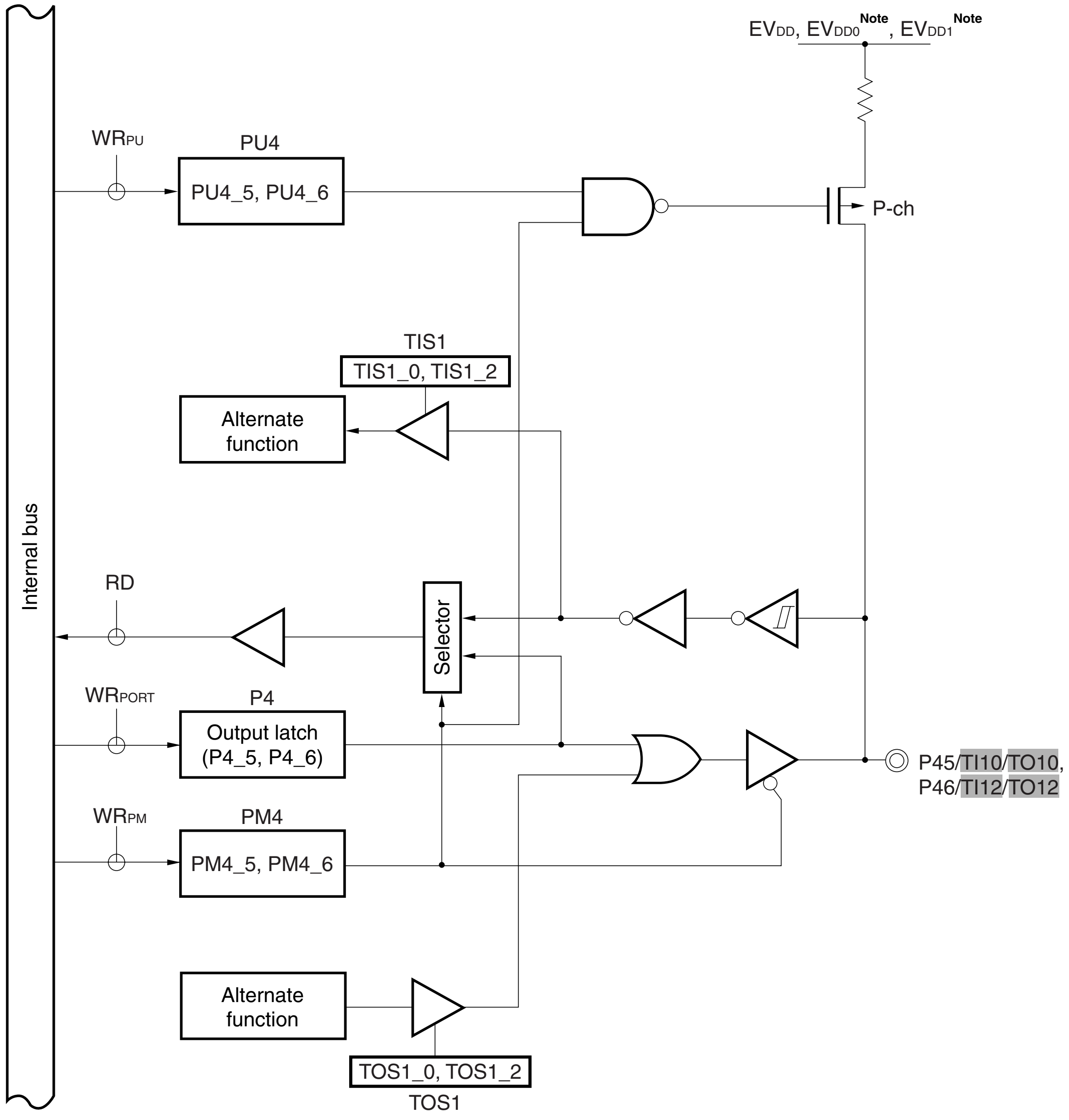


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P4:	Port register 4
PU4:	Pull-up resistor option register 4
PM4:	Port mode register 4
RD:	Read signal
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

Figure 4-33. Block Diagram of P45 and P46

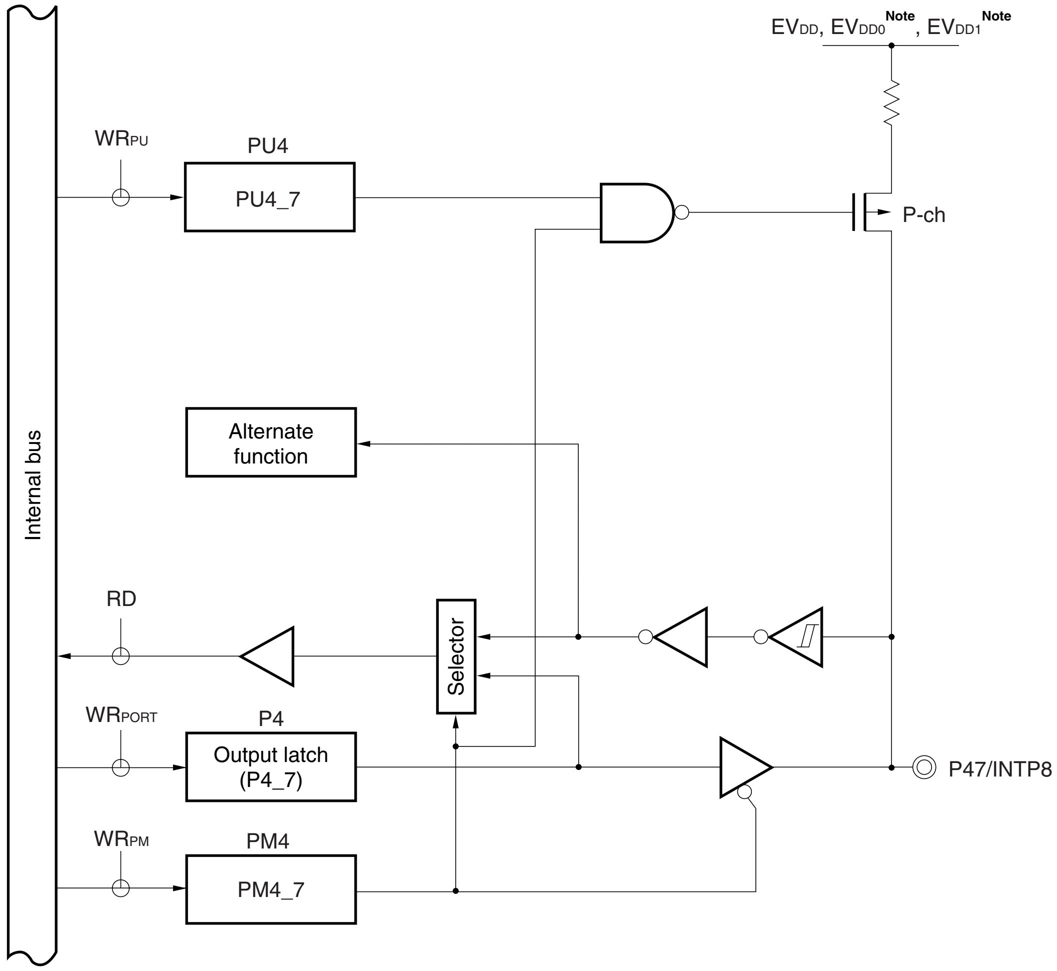


Note EV_{DD0}, EV_{DD1} are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P4:	Port register 4
PU4:	Pull-up resistor option register 4
PM4:	Port mode register 4
RD:	Read signal
TIS1:	Timer input select register 1
TOS1:	Timer output select register 1
WR××:	Write signal

Figure 4-34. Block Diagram of P47



Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

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/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P50/INTP3/TI20/TO20	–	√	√	√
P51/TI21/TO21	–	√	√	√
P52/TI22/STOPST/TO22	–	√	√	√
P53/TI23/TO23	–	√	√	√
P54/TI11/TO11	–	–	√	√
P55/TI13/TO13	–	–	√	√
P56/TI15/TO15	–	–	√	√
P57/TI17/TO17	–	–	√	√

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

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 external interrupt request input, timer I/O, and STOP status output.

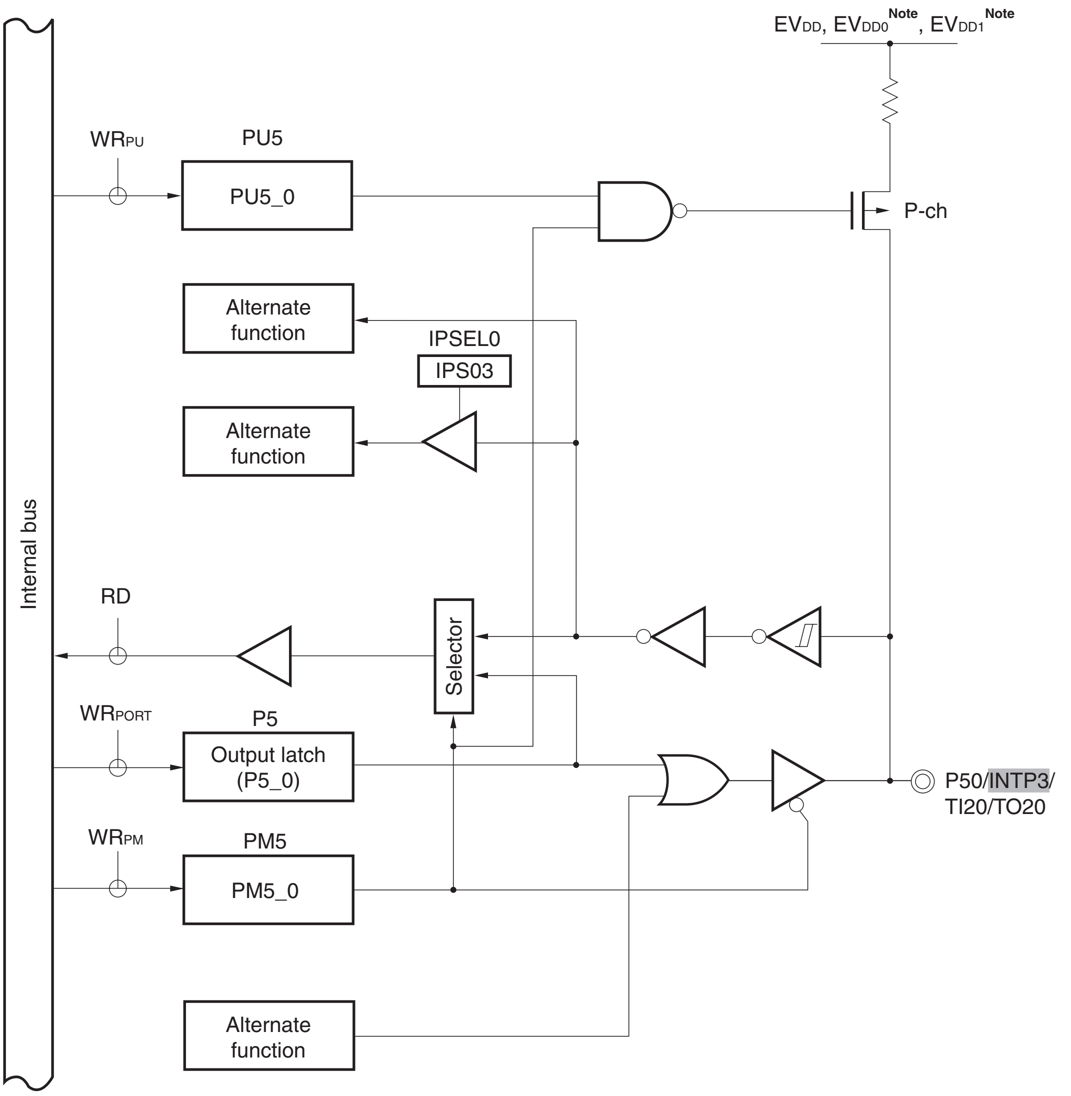
Reset signal generation sets port 5 to input mode.

Figures 4-35 to 4-39 show block diagrams of port 5.

Cautions 1. To use P50/INTP3/TI20/TO20, P51/TI21/TO21, P52/TI22/STOPST/TO22, P53/TI23/TO23, P54/TI11/TO11, P55/TI13/TO13, P56/TI15/TO15, or P57/TI17/TO17 as a general-purpose port, configure bits 1, 3, 5 and 7 (TO11, TO13, TO15, TO17) of timer output register 1 (TO1) and bits 0 to 3 (TO20 to TO23) of timer output register 2 (TO2) and bits 1, 3, 5 and 7 (TOE1_1, TOE1_3, TOE1_5, TOE1_7) of timer output enable register 1 (TOE1) and bits 0 to 3 (TOE2_0 to TOE2_3) of timer output enable register 2 (TOE2) to "0", which is the same as their default status setting.

2. The shaded pins are provided at two ports. Select either port by using the corresponding register.

Figure 4-35. Block Diagram of P50

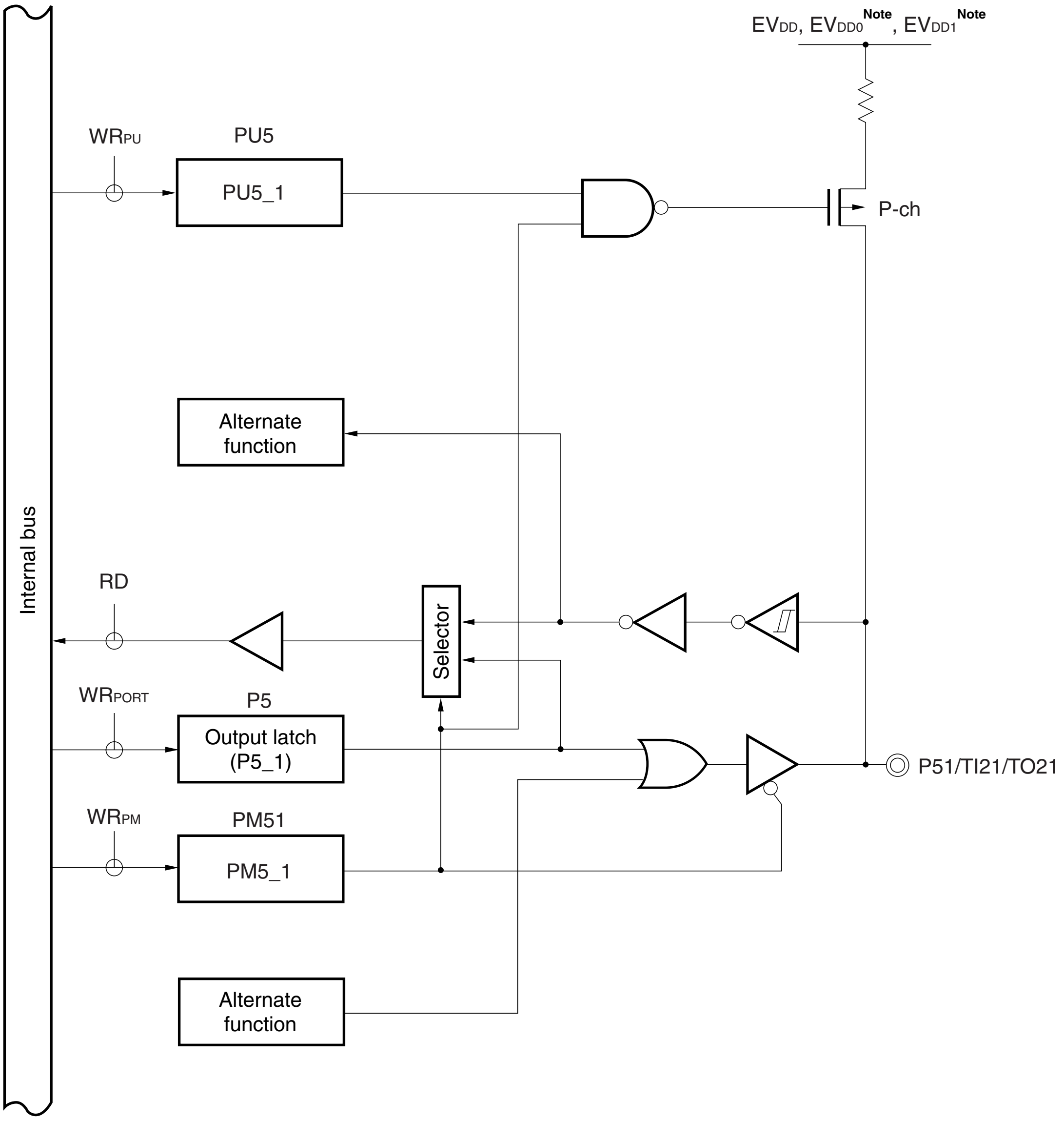


Note EVDD0, EVDD1 are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

IPSEL0:	External interrupt input pin selection register 0
P5:	Port register 5
PU5:	Pull-up resistor option register 5
PM5:	Port mode register 5
RD:	Read signal
WR××:	Write signal

Figure 4-36. Block Diagram of P51

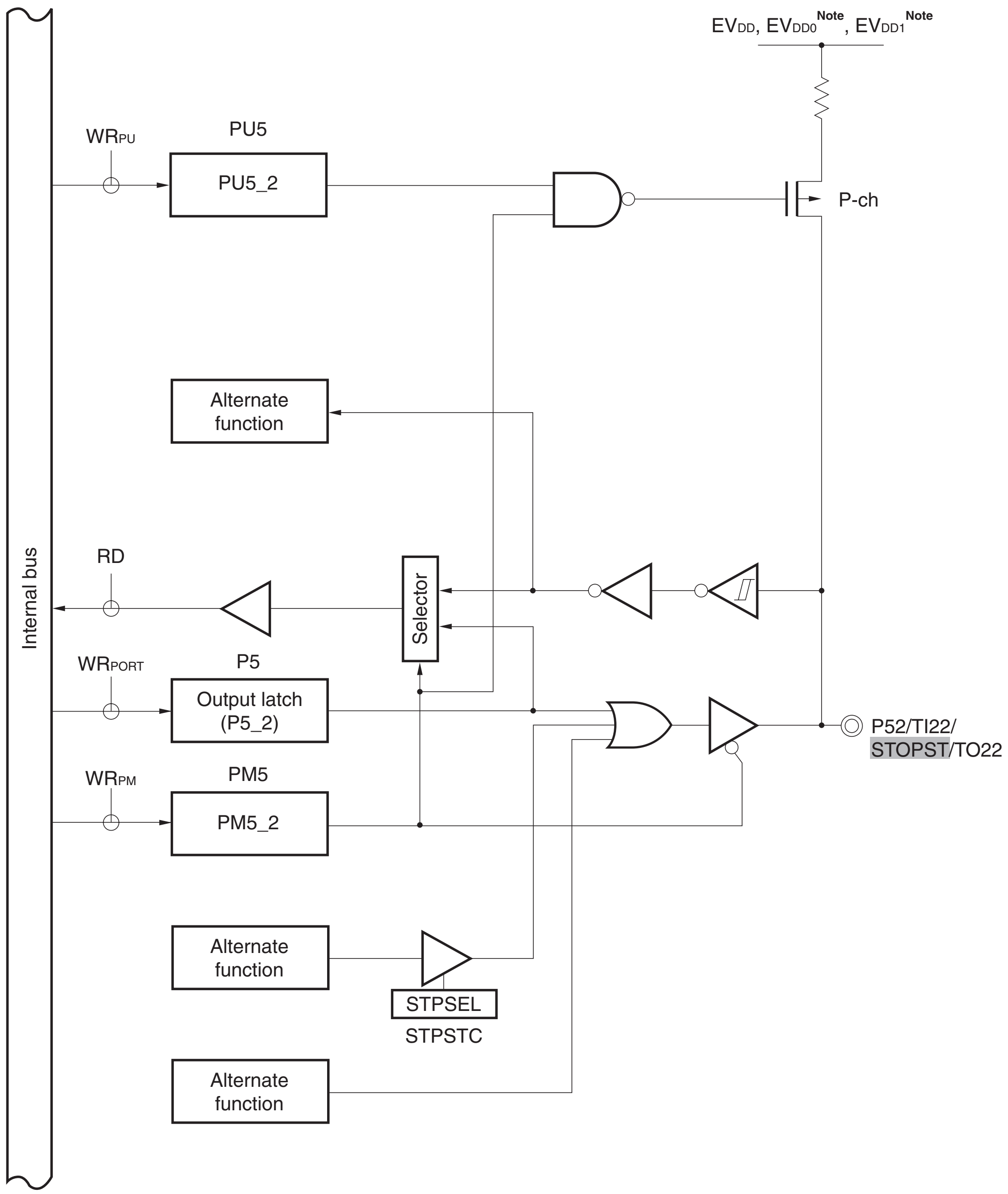


Note EV_{DD0}, EV_{DD1} are 78K0R/HG3 only

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

<R>

Figure 4-37. Block Diagram of P52

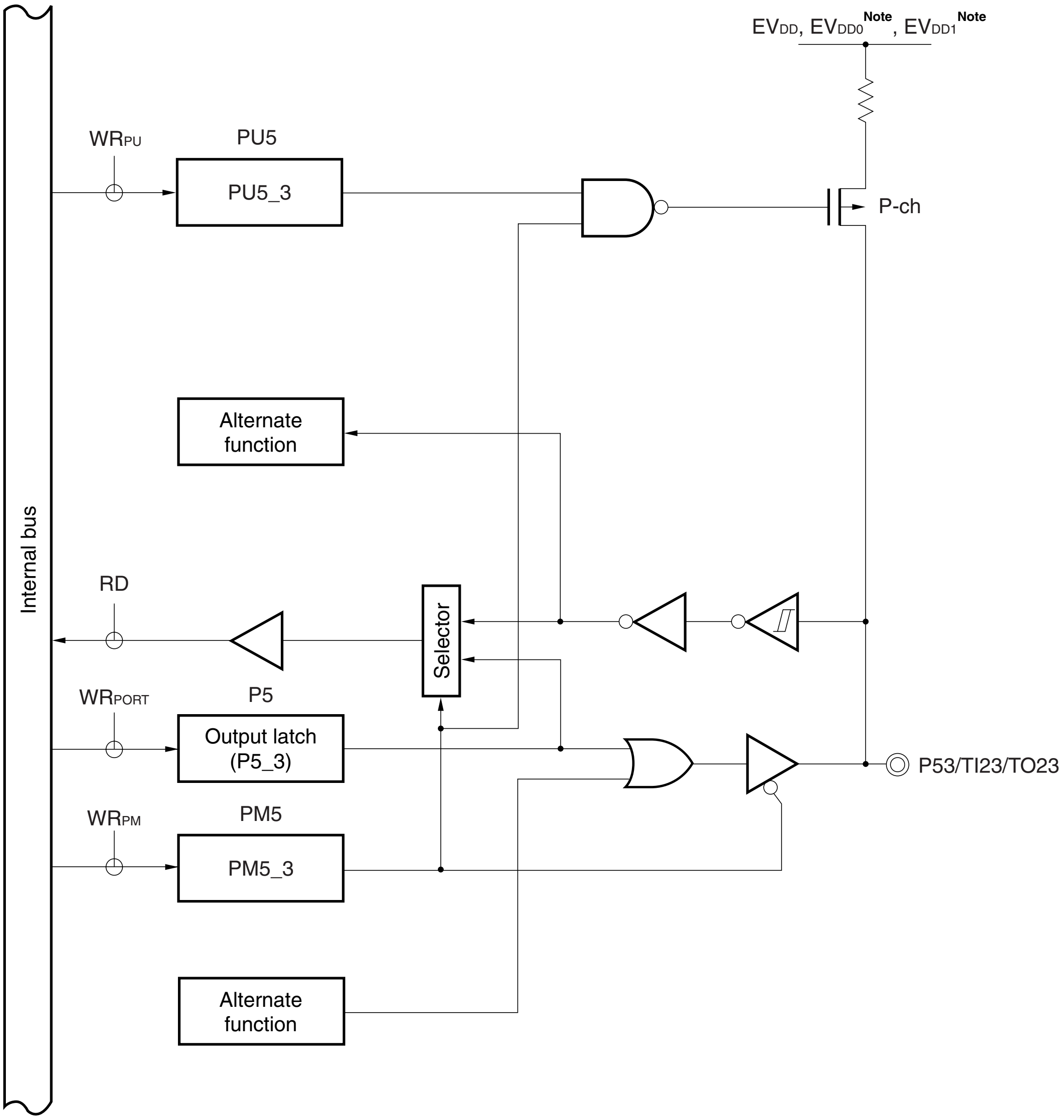


Note EVDD0, EVDD1 are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P5:	Port register 5
PU5:	Pull-up resistor option register 5
PM5:	Port mode register 5
RD:	Read signal
STPSTC:	STOP status output control register
WR××:	Write signal

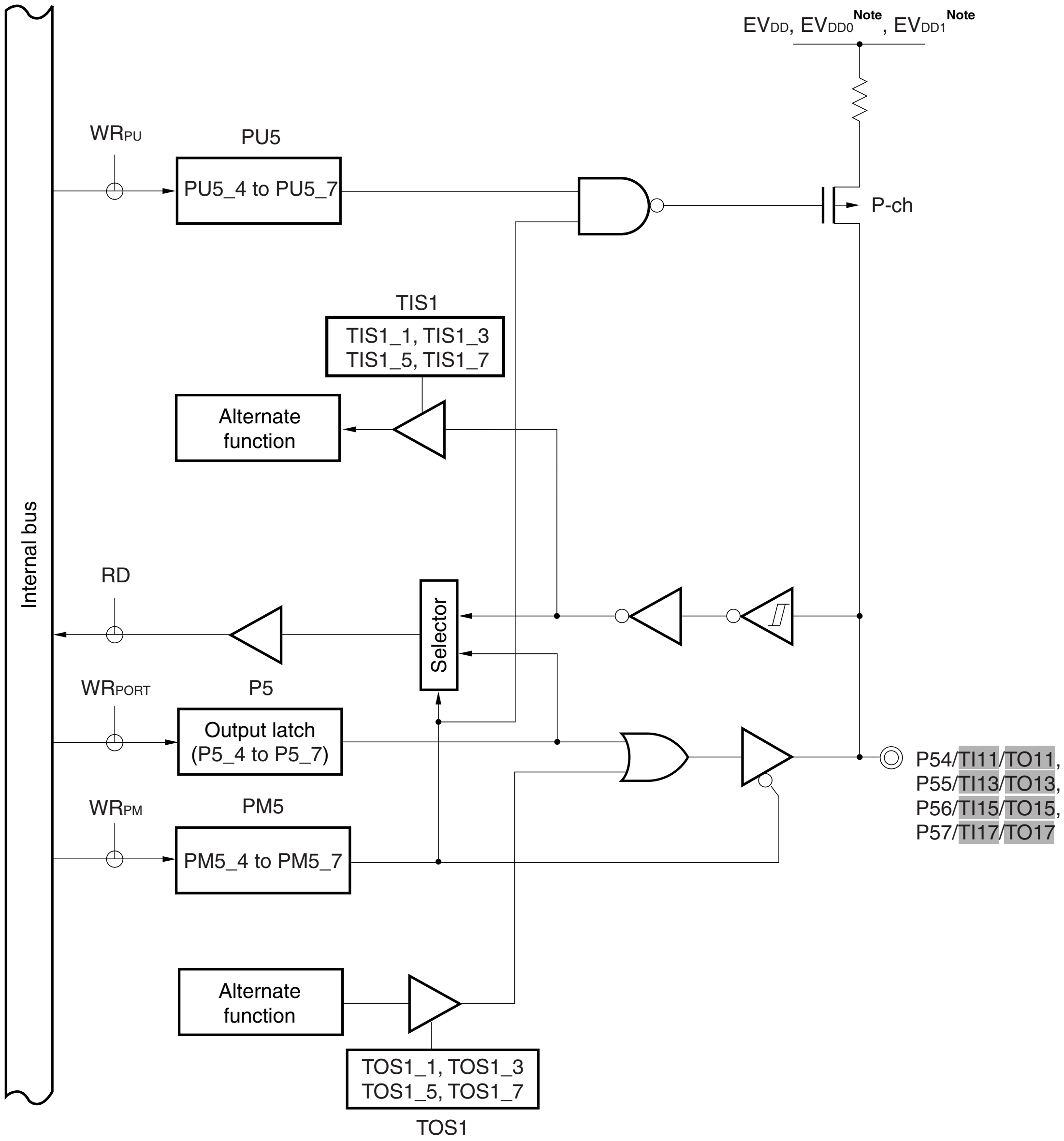
Figure 4-38. Block Diagram of P53



Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

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

Figure 4-39. Block Diagram of P54 to P57



Note EV_{DD0}, EV_{DD1} are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P5:	Port register 5
PU5:	Pull-up resistor option register 5
PM5:	Port mode register 5
RD:	Read signal
TIS1:	Timer input select register 1
TOS1:	Timer output select register 1
WR××:	Write signal

4.2.6 Port 6

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P60/SCK00/SCL11	√	√	√	√
P61/SI00/SDA11	√	√	√	√
P62/SO00	√	√	√	√
P63/SSI00	√	√	√	√
P64/TI14/TO14	–	–	√	√
P65/TI16/TO16	–	–	√	√
P66/TI00/TO00	–	–	√	√
P67/TI02/TO02	–	–	√	√

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

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). When the P60 to P67 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 6 (PU6).

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

Input to the P60, P61 and P63 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 6 (PIM6).

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

Reset signal generation sets port 6 to input mode.

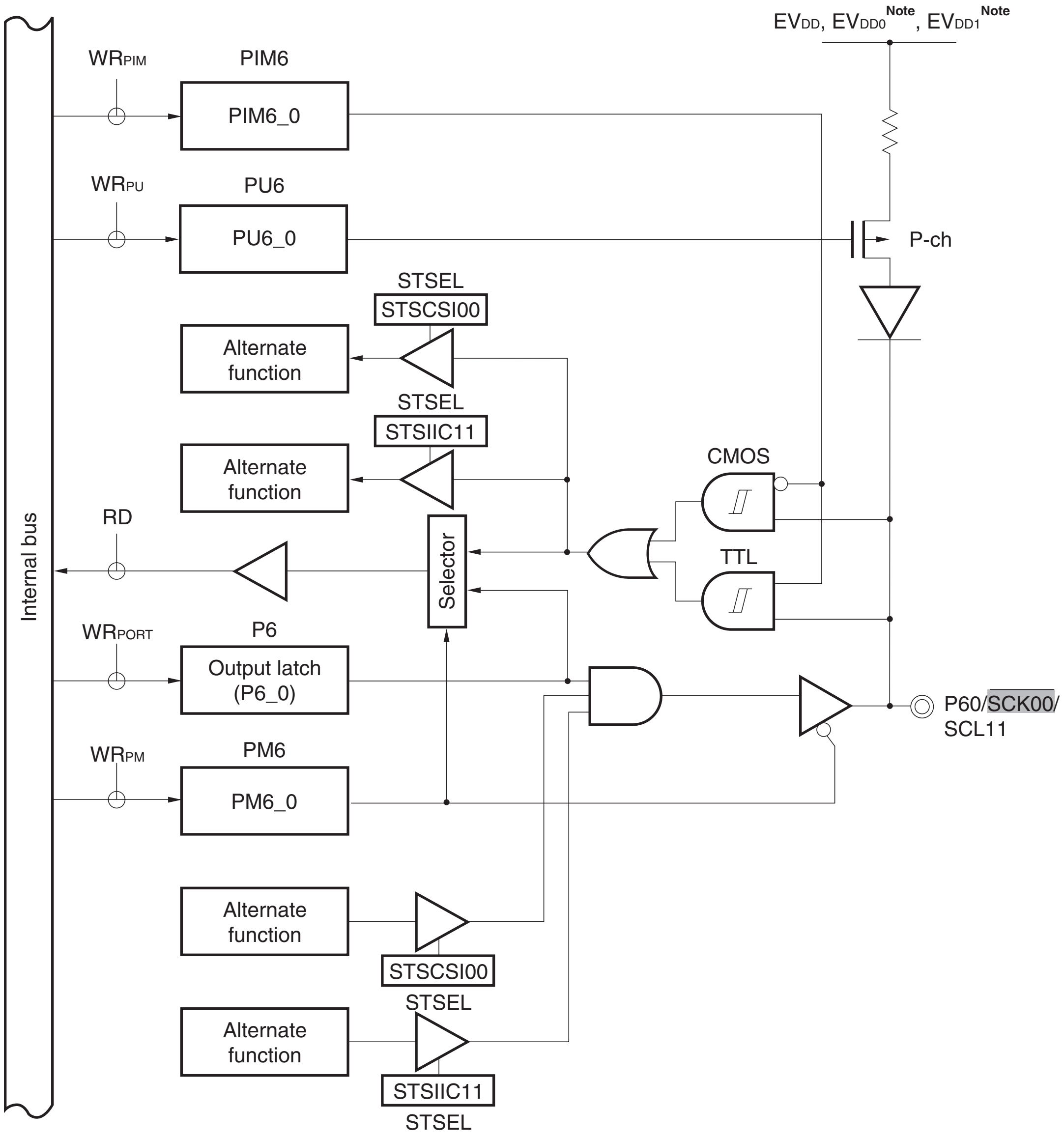
Figures 4-40 to 4-46 show block diagrams of port 6.

Cautions 1. To use P60/SCK00/SCL11, P61/SI00/SDA11, P62/SO00, or P63/SSI00 as a general-purpose port, note the serial array unit setting. For details, see Table 11-9 Relationship Between Register Settings and Pins (Channel 0 of Unit 0: CSI00, STSCSI00 = 1) and Table 11-12 Relationship Between Register Settings and Pins (Channel 1 of Unit 1: CSI11, IIC11).

2. To use P64/TI14/TO14, P65/TI16/TO16, P66/TI00/TO00, or P67/TI02/TO02 as a general-purpose port, configure bits 0 and 2 (TO00, TO02) of timer output register 0 (TO0) and bits 4 and 6 (TO14, TO16) of timer output register 1 (TO1) and bits 0 and 2 (TOE0_0, TOE0_2) of timer output enable register 0 (TOE0) and bits 4 and 6 (TOE1_4, TOE1_6) of timer output enable register 1 (TOE1) to "0", which is the same as their default status setting.

3. The shaded pins are provided at two ports. Select either port by using the corresponding register.

Figure 4-40. Block Diagram of P60

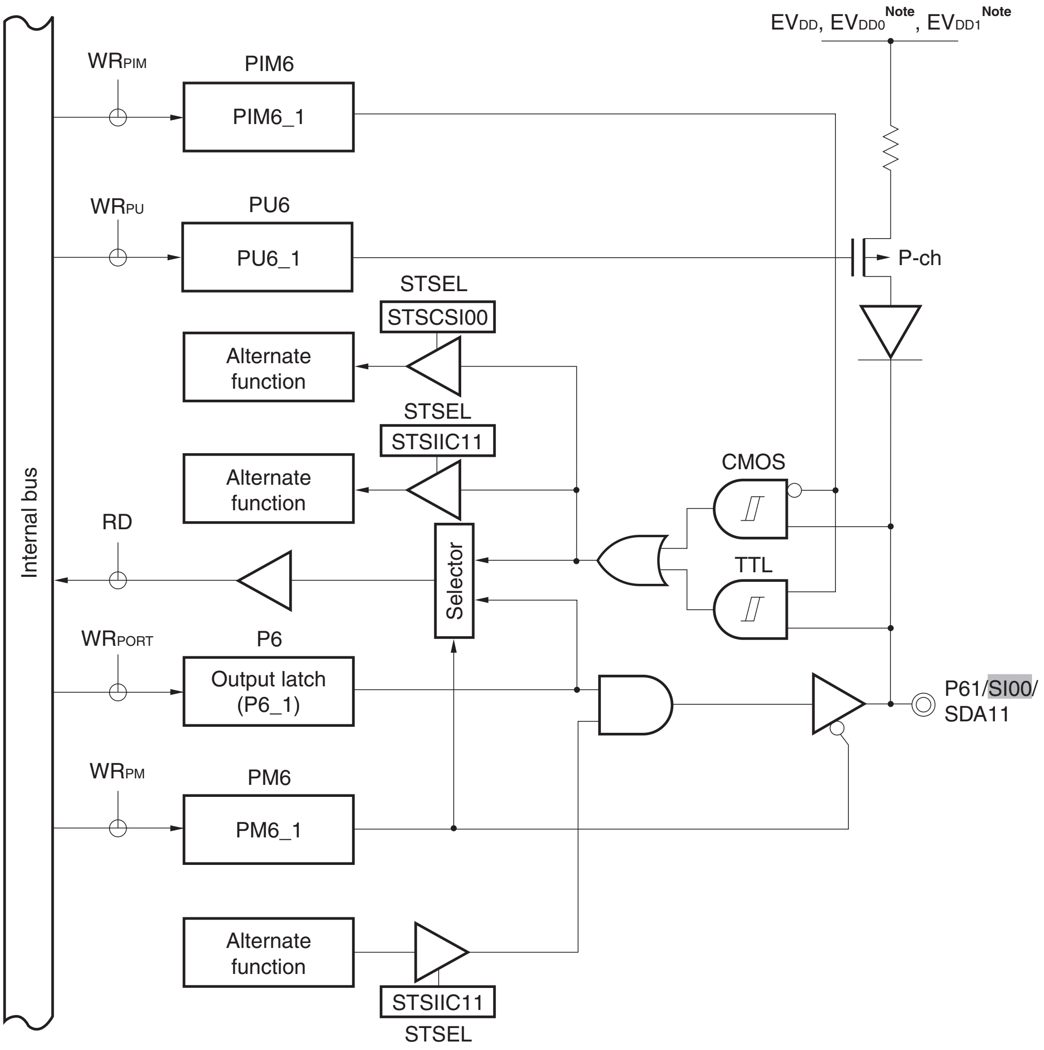


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P6:	Port register 6
PU6:	Pull-up resistor option register 6
PM6:	Port mode register 6
PIM6:	Port input mode register 6
RD:	Read signal
STSEL:	Serial communication pin select register
WR××:	Write signal

Figure 4-41. Block Diagram of P61

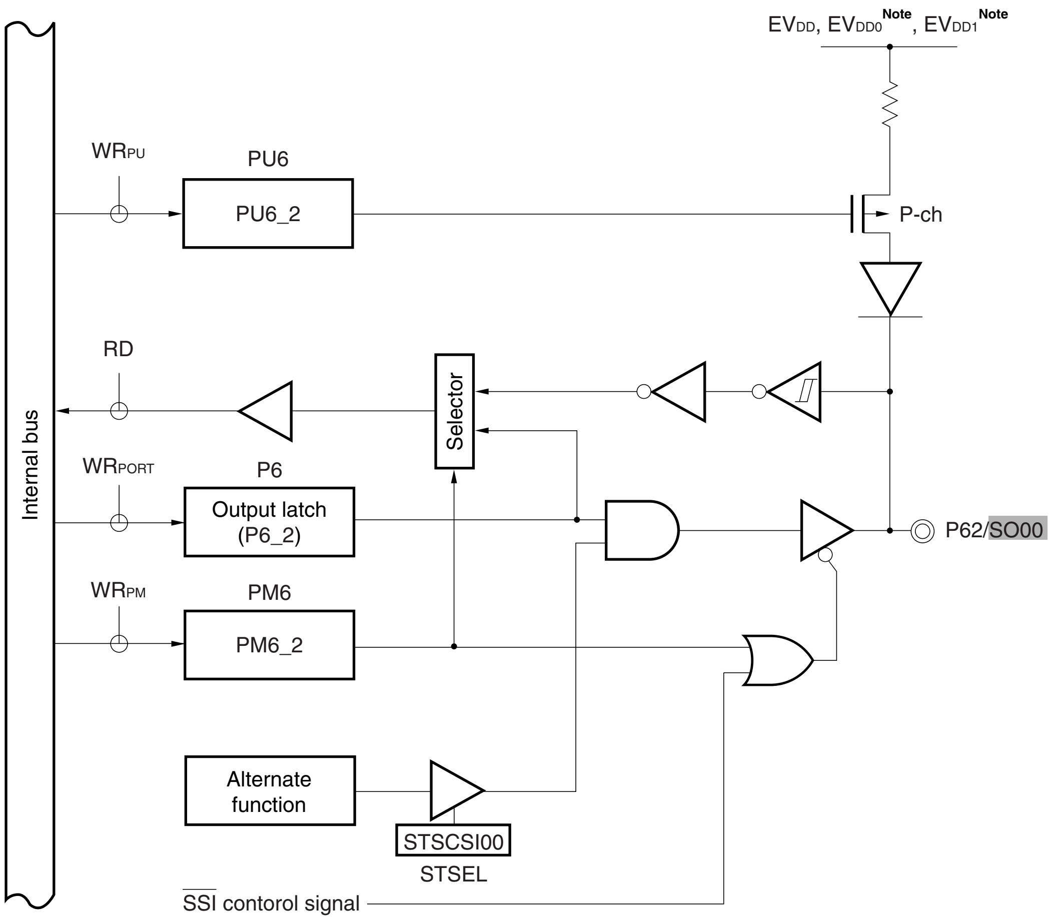


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P6:	Port register 6
PU6:	Pull-up resistor option register 6
PM6:	Port mode register 6
PIM6:	Port input mode register 6
RD:	Read signal
STSEL:	Serial communication pin select register
WR××:	Write signal

Figure 4-42. Block Diagram of P62

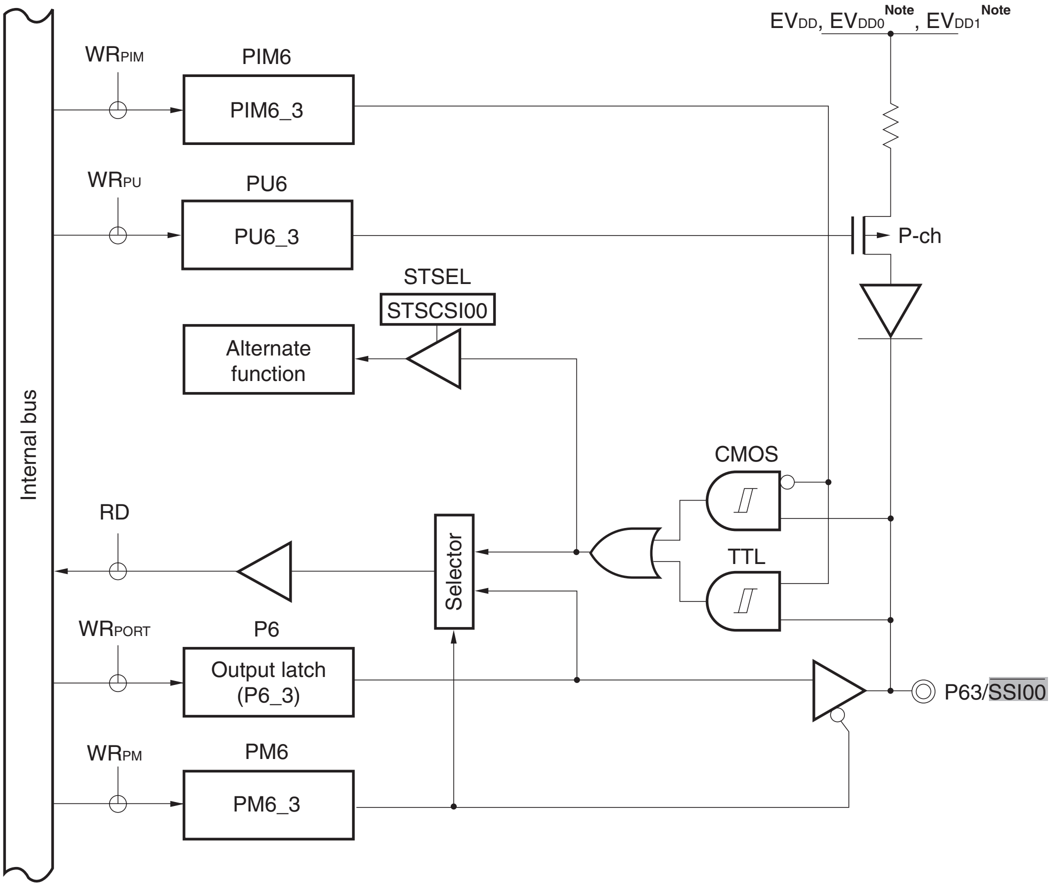


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P6:	Port register 6
PU6:	Pull-up resistor option register 6
PM6:	Port mode register 6
RD:	Read signal
STSEL:	Serial communication pin select register
WR××:	Write signal

Figure 4-43. Block Diagram of P63

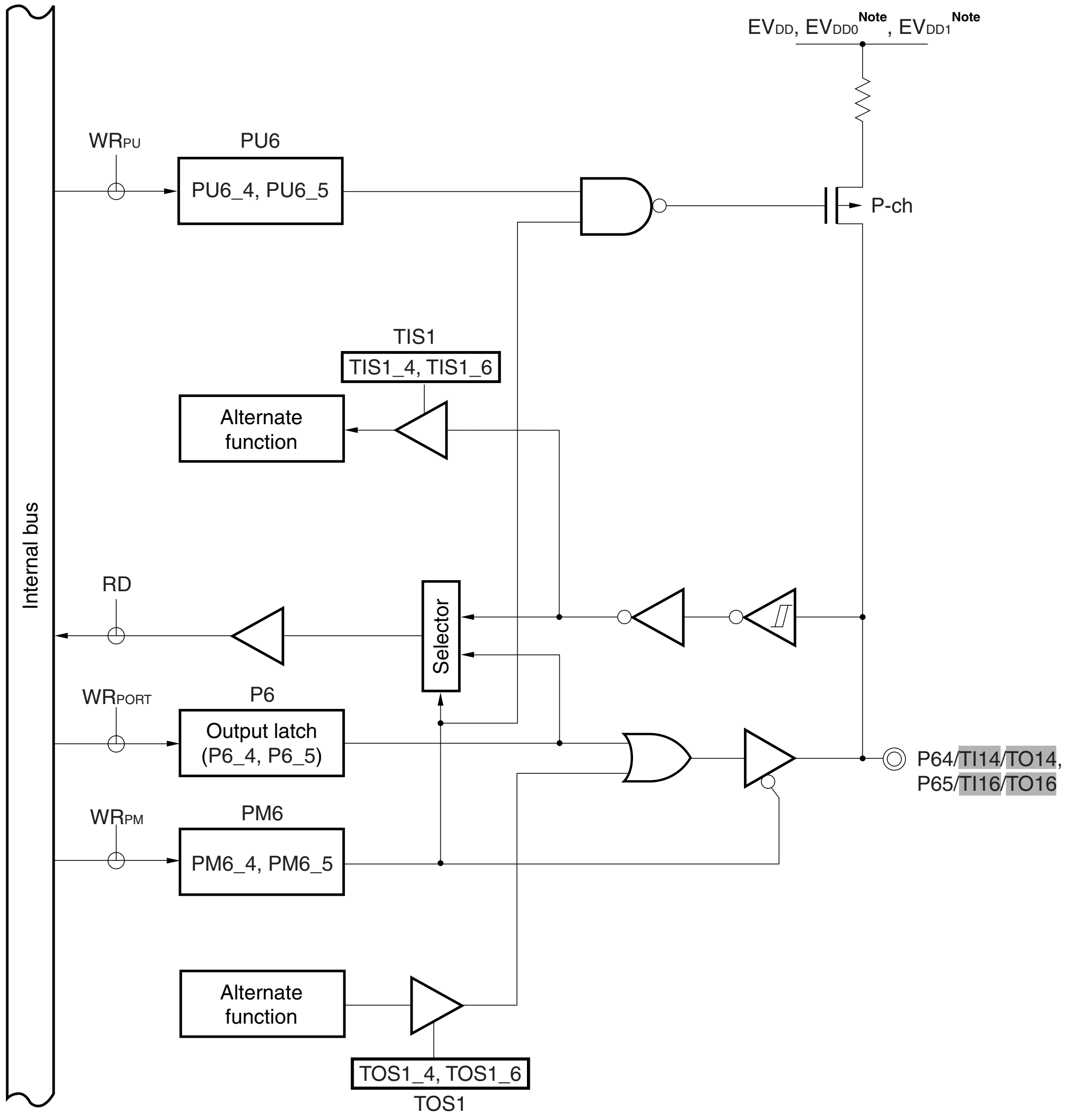


Note EV_{DD0}, EV_{DD1} are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P6:	Port register 6
PU6:	Pull-up resistor option register 6
PM6:	Port mode register 6
PIM6:	Port input mode register 6
RD:	Read signal
STSEL:	Serial communication pin select register
WR××:	Write signal

Figure 4-44. Block Diagram of P64 and P65

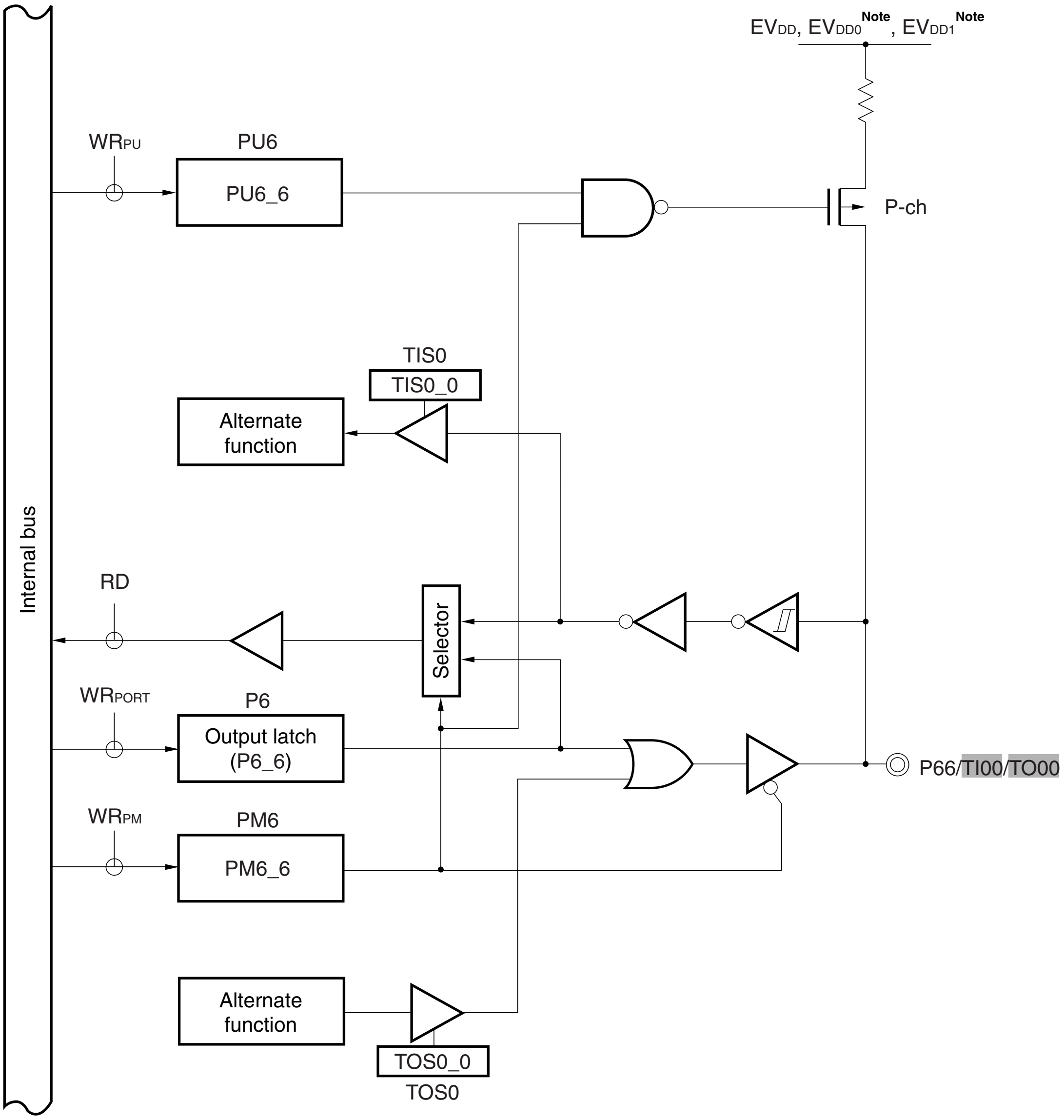


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P6:	Port register 6
PU6:	Pull-up resistor option register 6
PM6:	Port mode register 6
RD:	Read signal
TIS1:	Timer input select register 1
TOS1:	Timer output select register 1
WR××:	Write signal

Figure 4-45. Block Diagram of P66



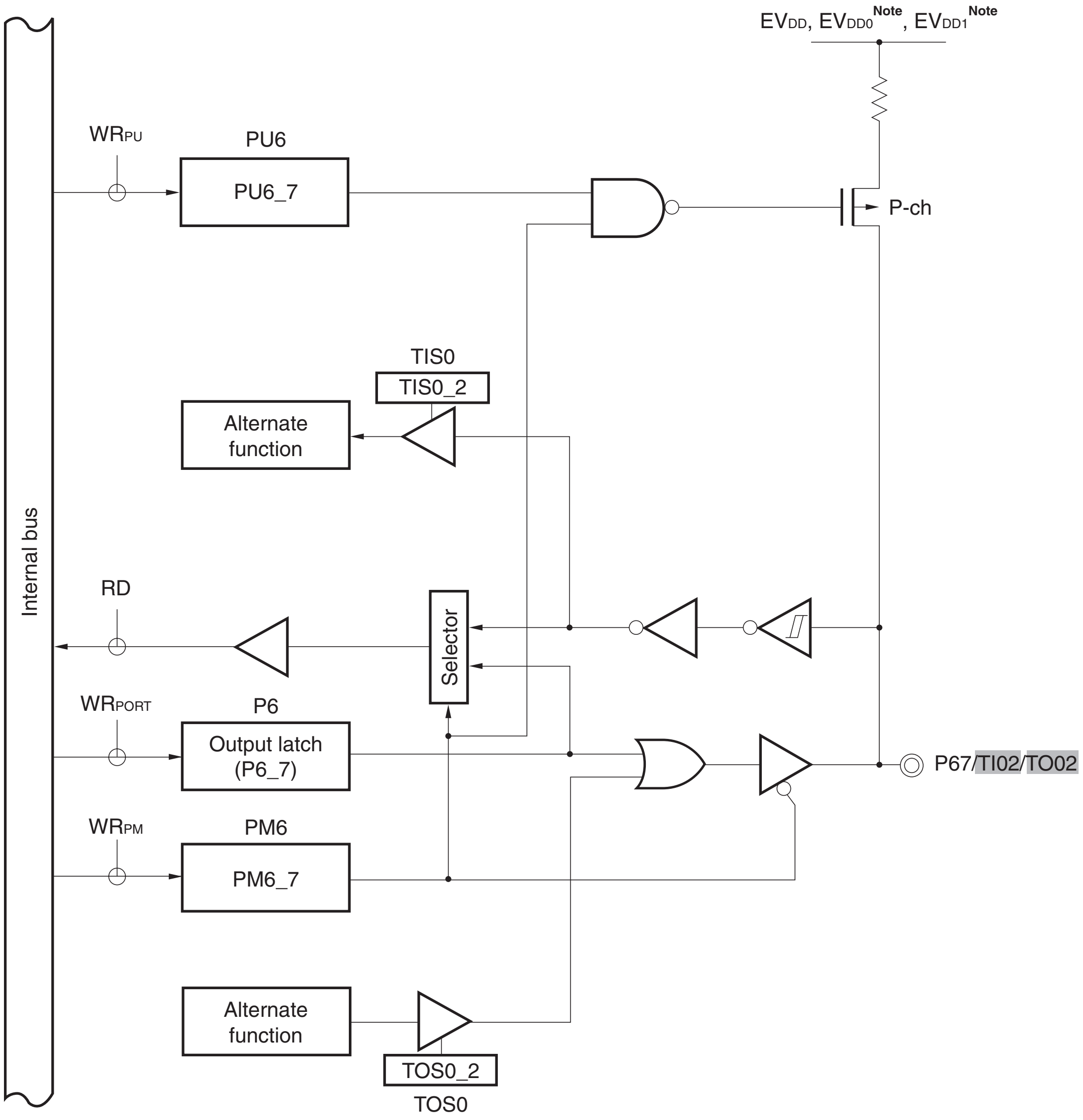
Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P6:	Port register 6
PU6:	Pull-up resistor option register 6
PM6:	Port mode register 6
RD:	Read signal
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

<R>

Figure 4-46. Block Diagram of P67



Note EVDD0, EVDD1 are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P6:	Port register 6
PU6:	Pull-up resistor option register 6
PM6:	Port mode register 6
RD:	Read signal
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

4.2.7 Port 7

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P70/INTP5/KR0/ TI15/TO15/LVIOUT	√	√	√	√
P71/INTP6/KR1/ TI17/TO17	√	√	√	√
P72/KR2/CTxD/ LTxD1	√	√	√	√
P73/KR3/CRxD/ LRxD1/INTPLR1	√	√	√	√
P74/KR4/SO01	–	√	√	√
P75/KR5/SI01	–	√	√	√
P76/KR6/$\overline{\text{SCK01}}$	–	√	√	√
P77/KR7/$\overline{\text{SSI01}}$	–	√	√	√

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

Port 7 is an I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 to P77 pins used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

Input to the P73, P75 to P77 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 7 (PIM7).

Output from the P72, P74 and P76 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 7 (POM7).

This port can also be used for external interrupt request input, key interrupt input, CAN data I/O, serial interface data I/O, clock I/O, timer I/O, and low-voltage detection flag output..

Reset signal generation sets port 7 to input mode.

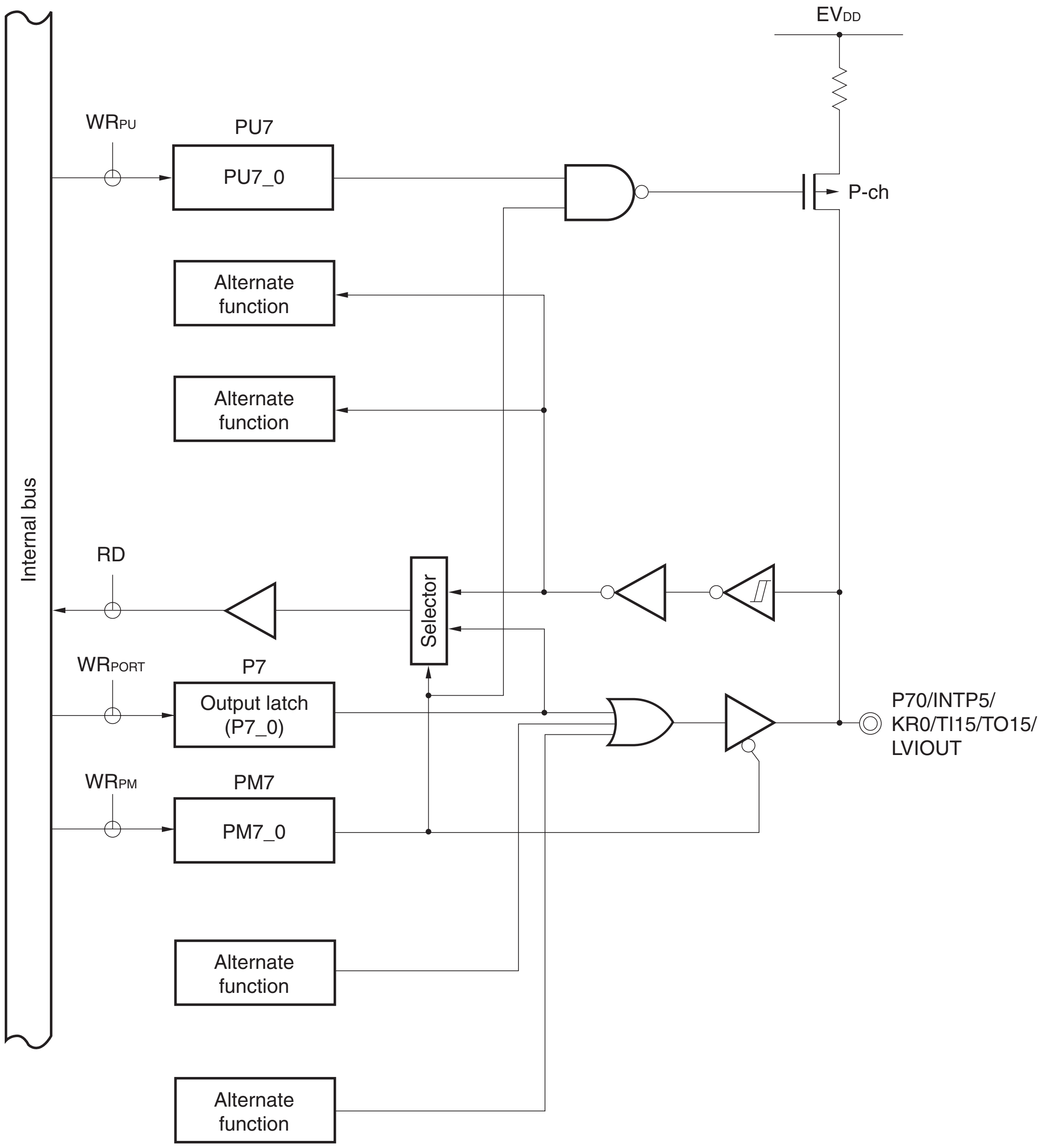
Figures 4-47 to 4-55 show a block diagram of port 7.

Cautions 1. To use P74/KR4/SO01, P75/KR5/SI01, P76/KR6/$\overline{\text{SCK01}}$, or P77/KR7/$\overline{\text{SSI01}}$ as a general-purpose port, note the serial array unit setting. For details, see Table 11-10 Relationship Between Register Settings and Pins (Channel 1 of Unit 0: CSI01).

2. To use P70/INTP5/KR0/TI15/TO15/LVIOUT, or P71/INTP6/KR1/TI17/TO17 as a general-purpose port, configure bits 5 and 7 (TO15, TO17) of timer output register 1 (TO1) and bits 5 and 7 (TOE1_5, TOE1_7) of timer output enable register 1 (TOE1) to "0", which is the same as their default status setting.

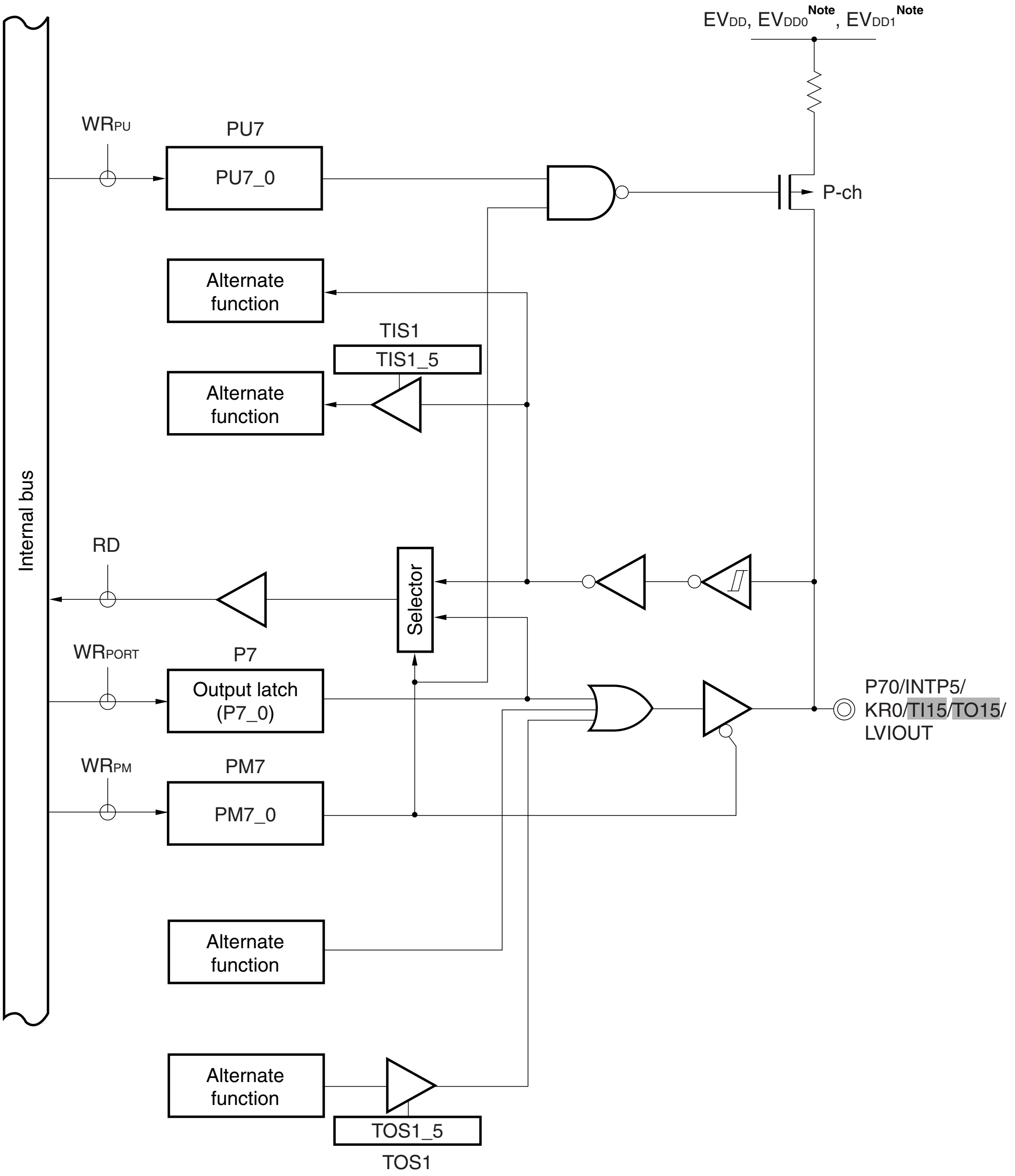
3. The shaded pins are provided at two ports. Select either port by using the corresponding register.

Figure 4-47. Block Diagram of P70 (78K0R/HC3, 78K0R/HE3)



P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
RD:	Read signal
WR××:	Write signal

Figure 4-48. Block Diagram of P70 (78K0R/HF3, 78K0R/HG3)

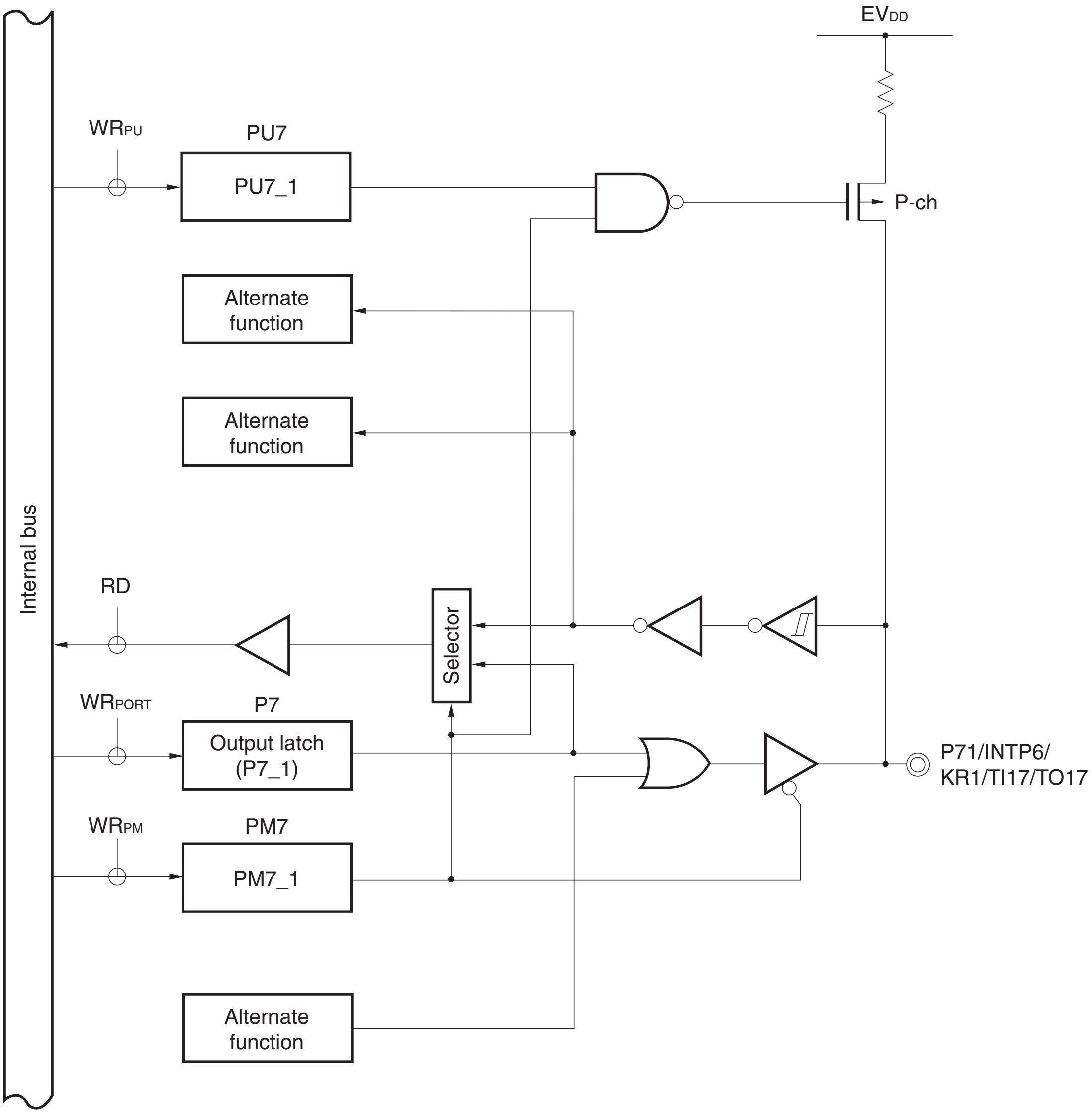


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

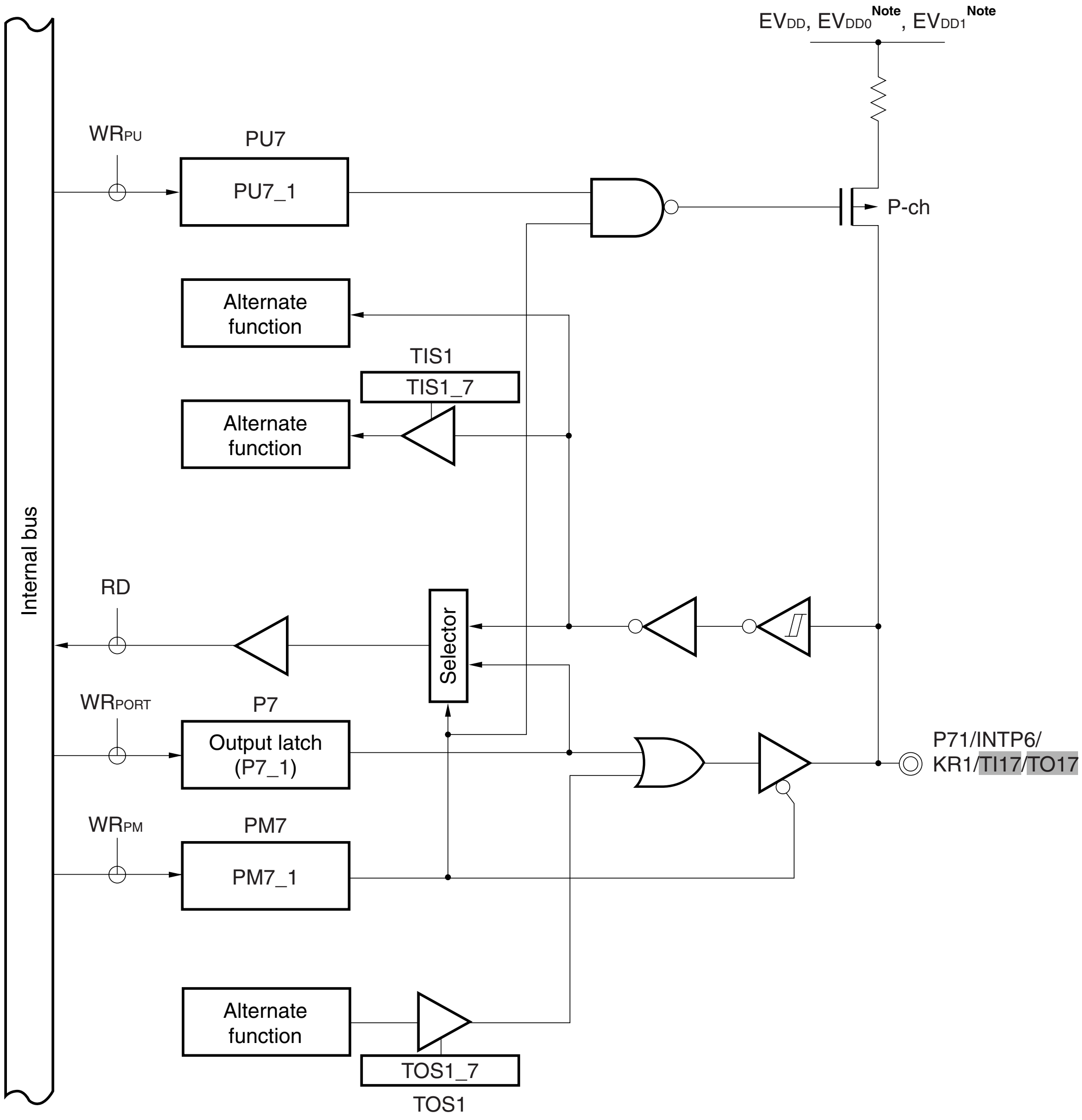
P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
RD:	Read signal
TIS1:	Timer input select register 1
TOS1:	Timer output select register 1
WR××:	Write signal

Figure 4-49. Block Diagram of P71 (78K0R/HC3, 78K0R/HE3)



P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
RD:	Read signal
TIS1:	Timer input select register 1
TOS1:	Timer output select register 1
WR××:	Write signal

Figure 4-50. Block Diagram of P71 (78K0R/HF3, 78K0R/HG3)

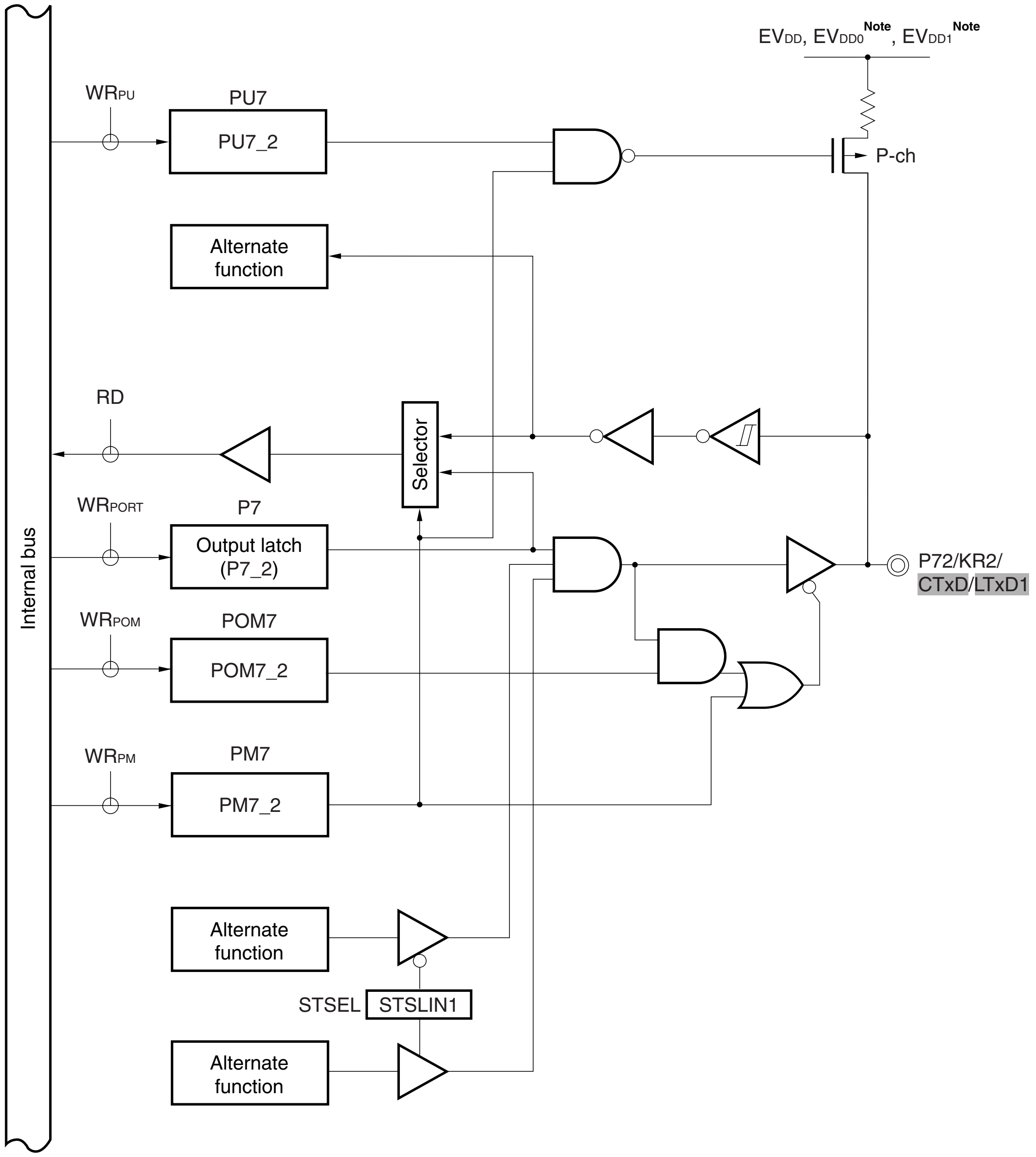


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
RD:	Read signal
TIS1:	Timer input select register 1
TOS1:	Timer output select register 1
WR××:	Write signal

Figure 4-51. Block Diagram of P72



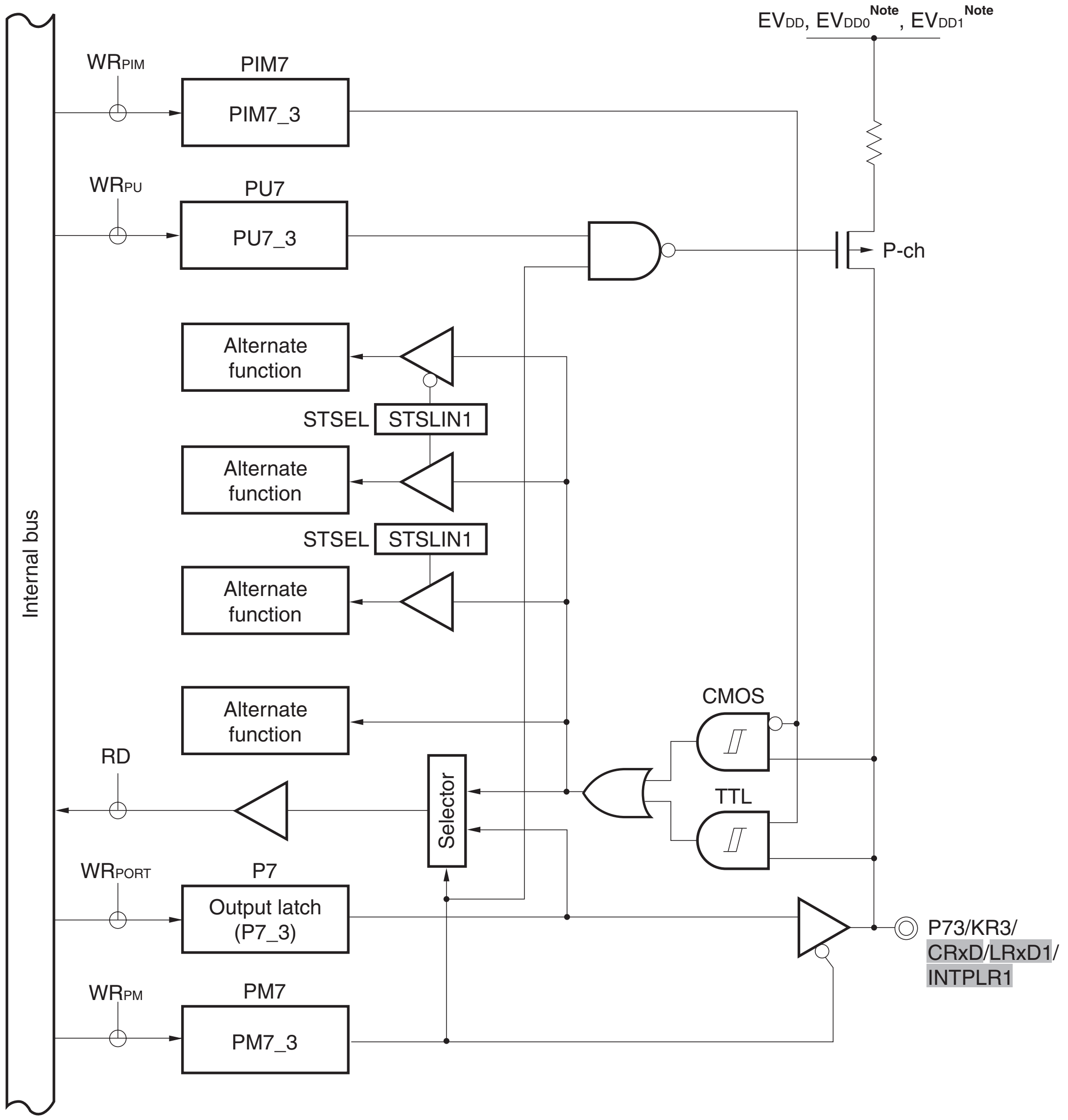
Note EVDD0, EVDD1 are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
POM7:	Port output mode register 7
RD:	Read signal
STSEL:	Serial communication pin select register
WR××:	Write signal

<R>

Figure 4-52. Block Diagram of P73

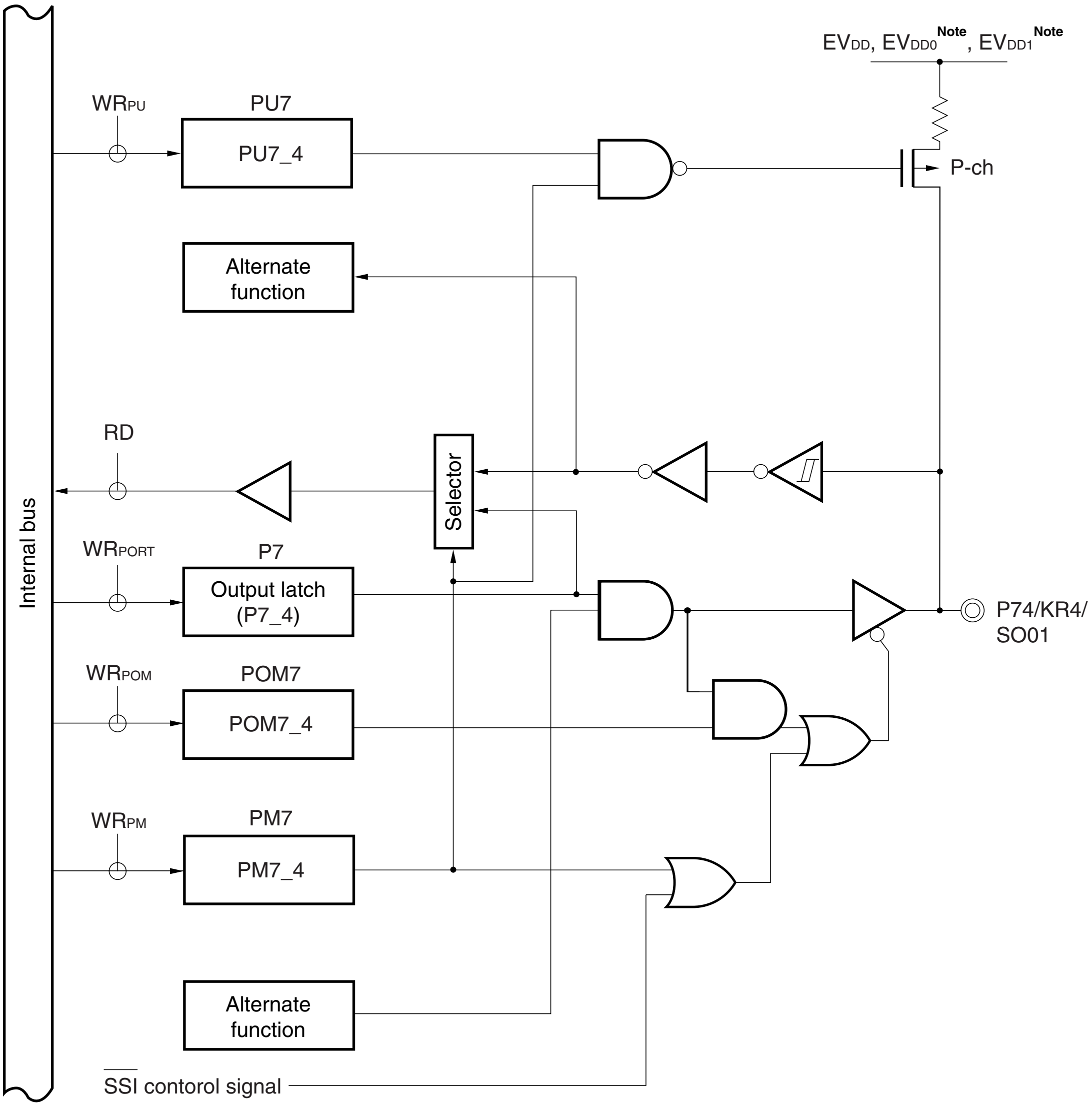


Note EVDD0, EVDD1 are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
PIM7:	Port input mode register 7
RD:	Read signal
STSEL:	Serial communication pin select register
WR××:	Write signal

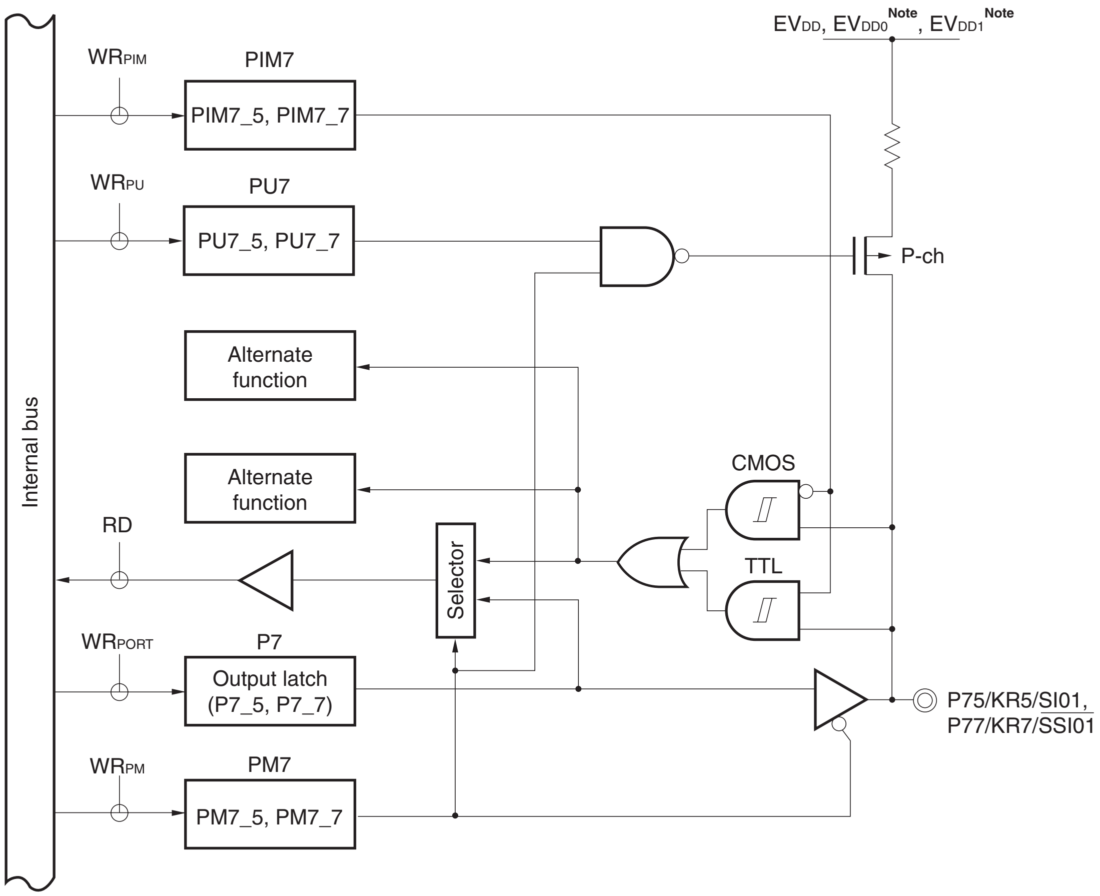
Figure 4-53. Block Diagram of P74



Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
POM7:	Port output mode register 7
RD:	Read signal
WR××:	Write signal

Figure 4-54. Block Diagram of P75 and P77

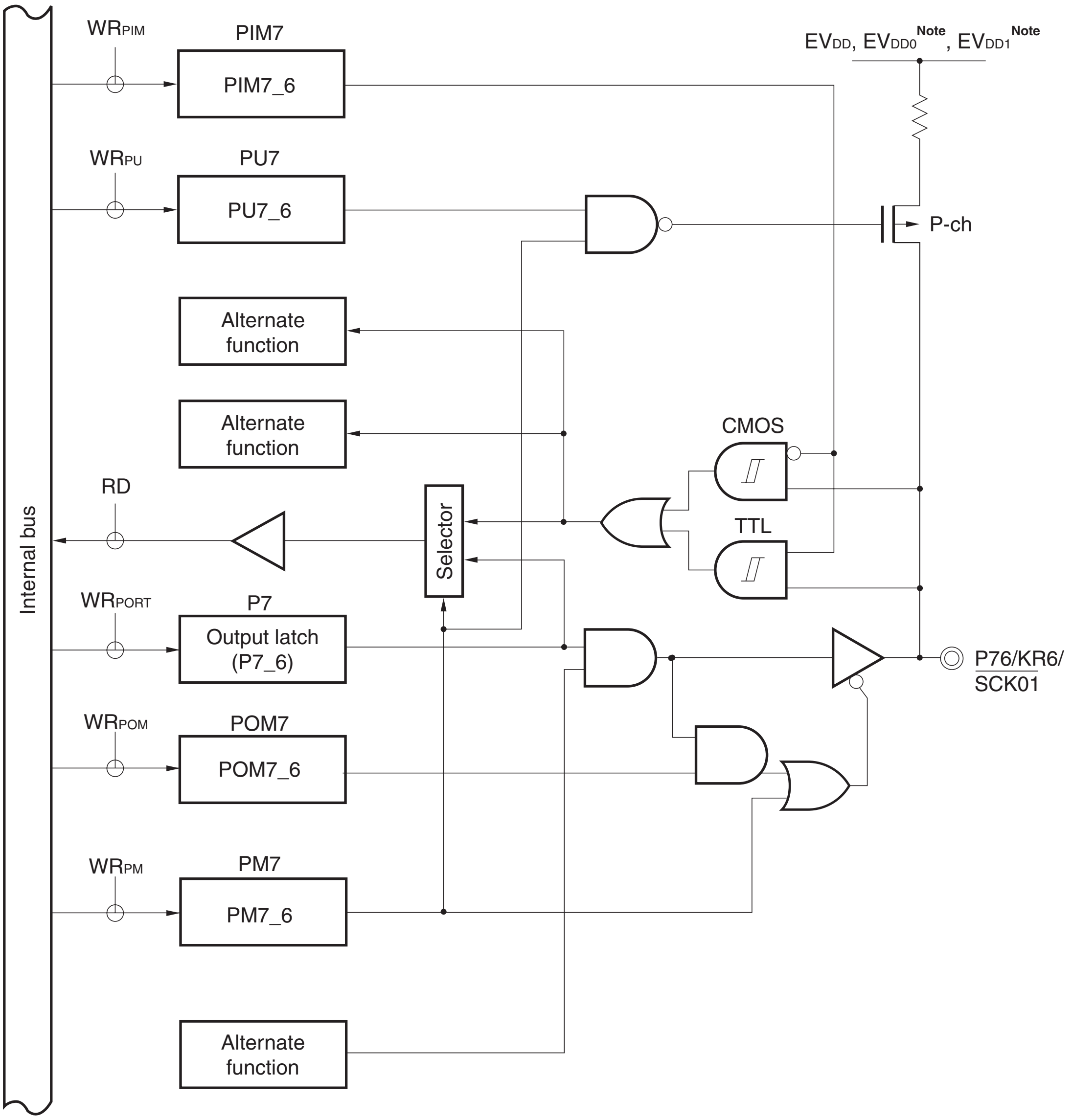


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
PIM7:	Port input mode register 7
RD:	Read signal
WR××:	Write signal

<R>

Figure 4-55. Block Diagram of P76



Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
PIM7:	Port input mode register 7
POM7:	Port output mode register 7
RD:	Read signal
WR××:	Write signal

4.2.8 Port 8

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P80/ANI00	√			
P81/ANI01	√			
P82/ANI02	√			
P83/ANI03	√			
P84/ANI04	√			
P85/ANI05	√			
P86/ANI06	√			
P87/ANI07	√			

Remark √: Mounted

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).

This port can also be used for A/D converter analog input.

To use P80/ANI00 to P87/ANI07 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 PM8. Use these pins starting from the upper bit.

To use P80/ANI00 to P87/ANI07 as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM8.

To use P80/ANI00 to P87/ANI07 as analog input pins, set them in the analog input mode by using the A/D port configuration register (ADPC) and in the input mode by using PM8. Use these pins starting from the lower bit.

Table 4-5. Setting Functions of P80/ANI00 to P87/ANI07 Pins

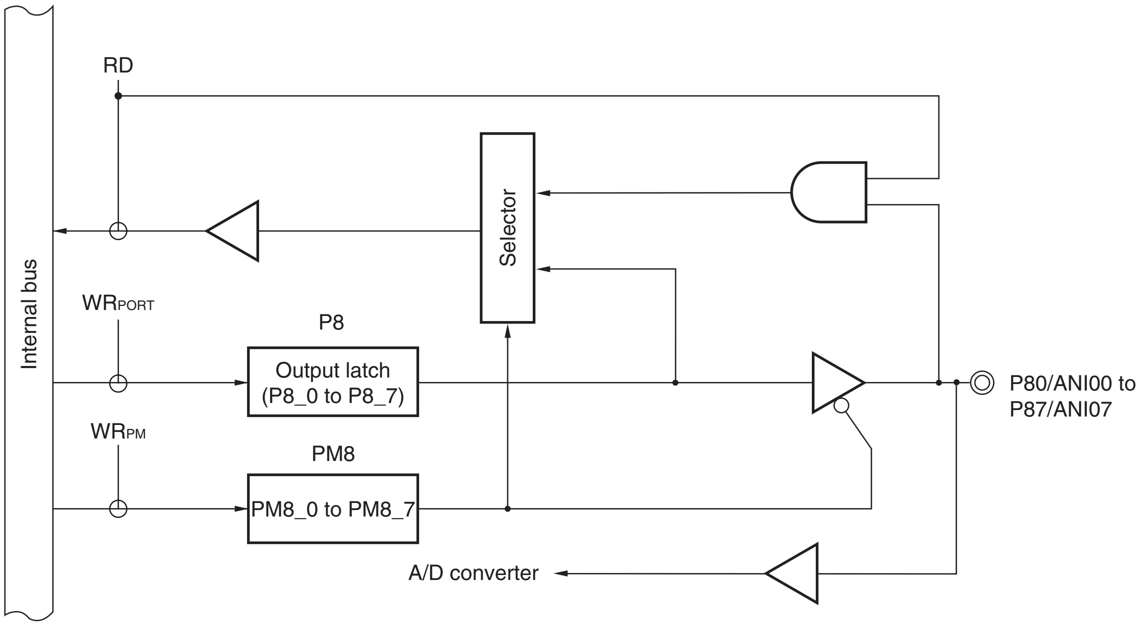
ADPC	PM8	ADS	P80/ANI00 to P87/ANI07 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P80/ANI00 to P87/ANI07 are set in the digital input mode when the reset signal is generated.

Figure 4-56 shows a block diagram of port 8.

Caution **Make the AV_{REF} pin the same potential as the V_{DD} pin when using port 8 as a digital port.**
When using port 8 as a digital input port, set port mode register 8, and then dummy-read the port register. When not specifying a setting for port mode register 8, dummy-read the port register after reset and before using the port register.

Figure 4-56. Block Diagram of P80 to P87



P8:	Port register 8
PM8:	Port mode register 8
RD:	Read signal
WR××:	Write signal

4.2.9 Port 9

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P90/ANI08	√	√	√	√
P91/ANI09	√	√	√	√
P92/ANI10	√	√	√	√
P93/ANI11	–	√	√	√
P94/ANI12	–	√	√	√
P95/ANI13	–	√	√	√
P96/ANI14	–	√	√	√
P97/ANI15	–	–	√	√

Remark √: Mounted, –: Not mounted

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).

This port can also be used for A/D converter analog input.

To use P90/ANI08 to P97/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 PM9. Use these pins starting from the upper bit.

To use P90/ANI08 to P97/ANI15 as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM9.

To use P90/ANI08 to P97/ANI15 as analog input pins, set them in the analog input mode by using the A/D port configuration register (ADPC) and in the input mode by using PM9. Use these pins starting from the lower bit.

Table 4-6. Setting Functions of P90/ANI08 to P97/ANI15 Pins

ADPC	PM9	ADS	P90/ANI08 to P97/ANI15 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

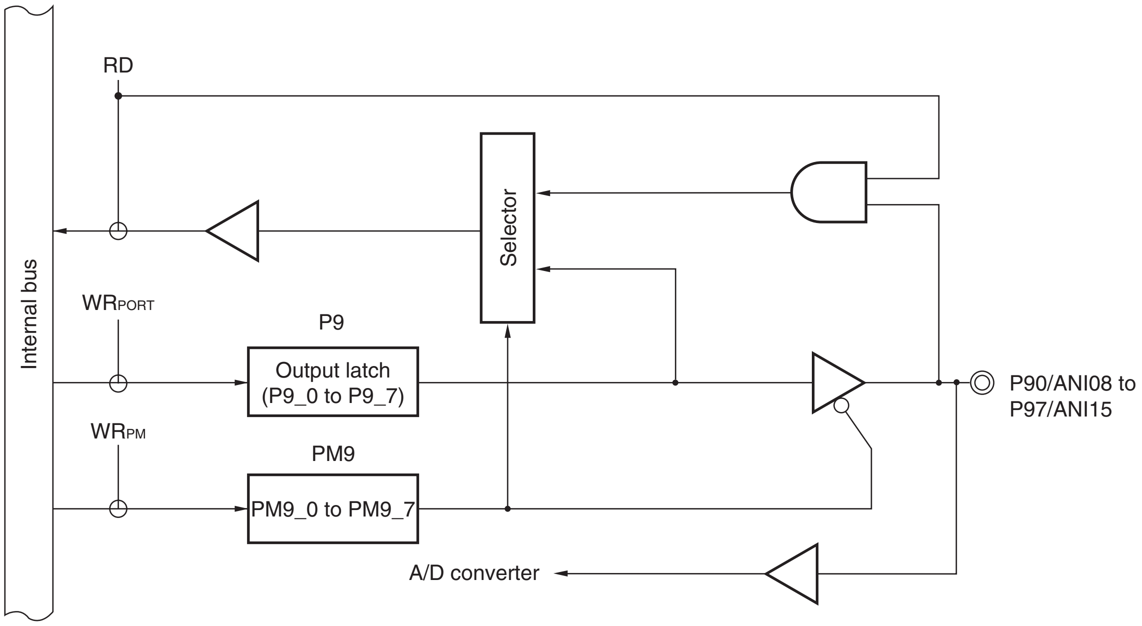
All P90/ANI08 to P97/ANI15 are set in the digital input mode when the reset signal is generated.

Figure 4-57 shows a block diagram of port 9.

Caution Make the AV$_{REF}$ pin the same potential as the V$_{DD}$ pin when using port 9 as a digital port.

When using port 9 as a digital input port, set port mode register 9, and then dummy-read the port register. When not specifying a setting for port mode register 9, dummy-read the port register after reset and before using the port register.

Figure 4-57. Block Diagram of P90 to P97



P9:	Port register 9
PM9:	Port mode register 9
RD:	Read signal
WR××:	Write signal

4.2.10 Port 10

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P100/ANI16	–	–	–	√
P101/ANI17	–	–	–	√
P102/ANI18	–	–	–	√
P103/ANI19	–	–	–	√
P104/ANI20	–	–	–	√
P105/ANI21	–	–	–	√
P106/ANI22	–	–	–	√
P107/ANI23	–	–	–	√

Remark √: Mounted, –: Not mounted

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).

This port can also be used for A/D converter analog input.

To use P100/ANI16 to P107/ANI23 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 PM10. Use these pins starting from the upper bit.

To use P100/ANI16 to P107/ANI23 as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM10.

To use P100/ANI16 to P107/ANI23 as analog input pins, set them in the analog input mode by using the A/D port configuration register (ADPC) and in the input mode by using PM10. Use these pins starting from the lower bit.

Table 4-5. Setting Functions of P100/ANI16 to P107/ANI23 Pins

ADPC	PM10	ADS	P100/ANI16 to P107/ANI23 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

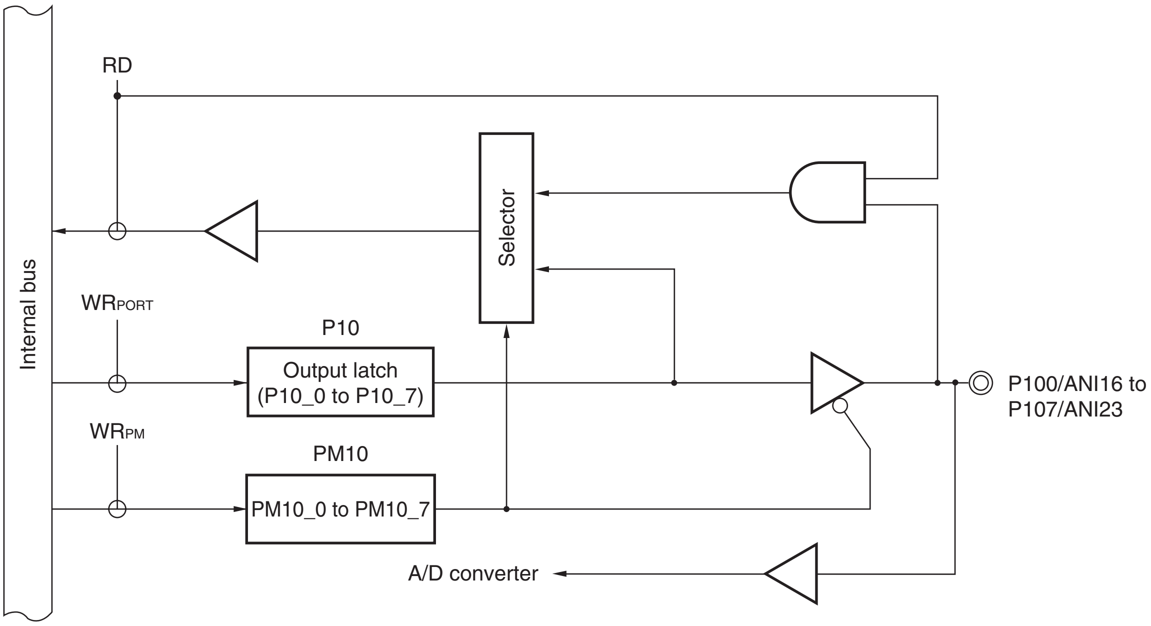
All P100/ANI16 to P107/ANI23 are set in the digital input mode when the reset signal is generated.

Figure 4-58 shows a block diagram of port 10.

Caution Make the AV$_{REF}$ pin the same potential as the V$_{DD}$ pin when using port 10 as a digital port.

When using port 10 as a digital input port, set port mode register 10, and then dummy-read the port register. When not specifying a setting for port mode register 10, dummy-read the port register after reset and before using the port register.

Figure 4-58. Block Diagram of P100 to P107



P10:	Port register 10
PM10:	Port mode register 10
RD:	Read signal
WR××:	Write signal

4.2.11 Port 12

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P120/INTP0/EXLVI/TI11/TO11	√	√	**Note**	**Note**
P121/X1	√	√	√	√
P122/X2/EXCLK	√	√	√	√
P123	√	√	√	√
P124/EXCLKS	√	√	√	√
P125/INTP1/ADTRG/TI03/TO03	√	√	√	√
P126/TI01/TO01	–	–	√	√
P127/TI03/TO03	–	–	–	√

Note TI11, TO11 pins are not mounted. Port functions other than TI11 and TO11 as well as shared functions are provided.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark √: Mounted, –: Not mounted

P120 and P125 to P127 are an I/O port with an output latch. P120 and P125 to P127 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

P121 to P124 are 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, external clock input for subclock, external trigger input for A/D converter, and timer I/O.

Reset signal generation sets port 12 to input mode.

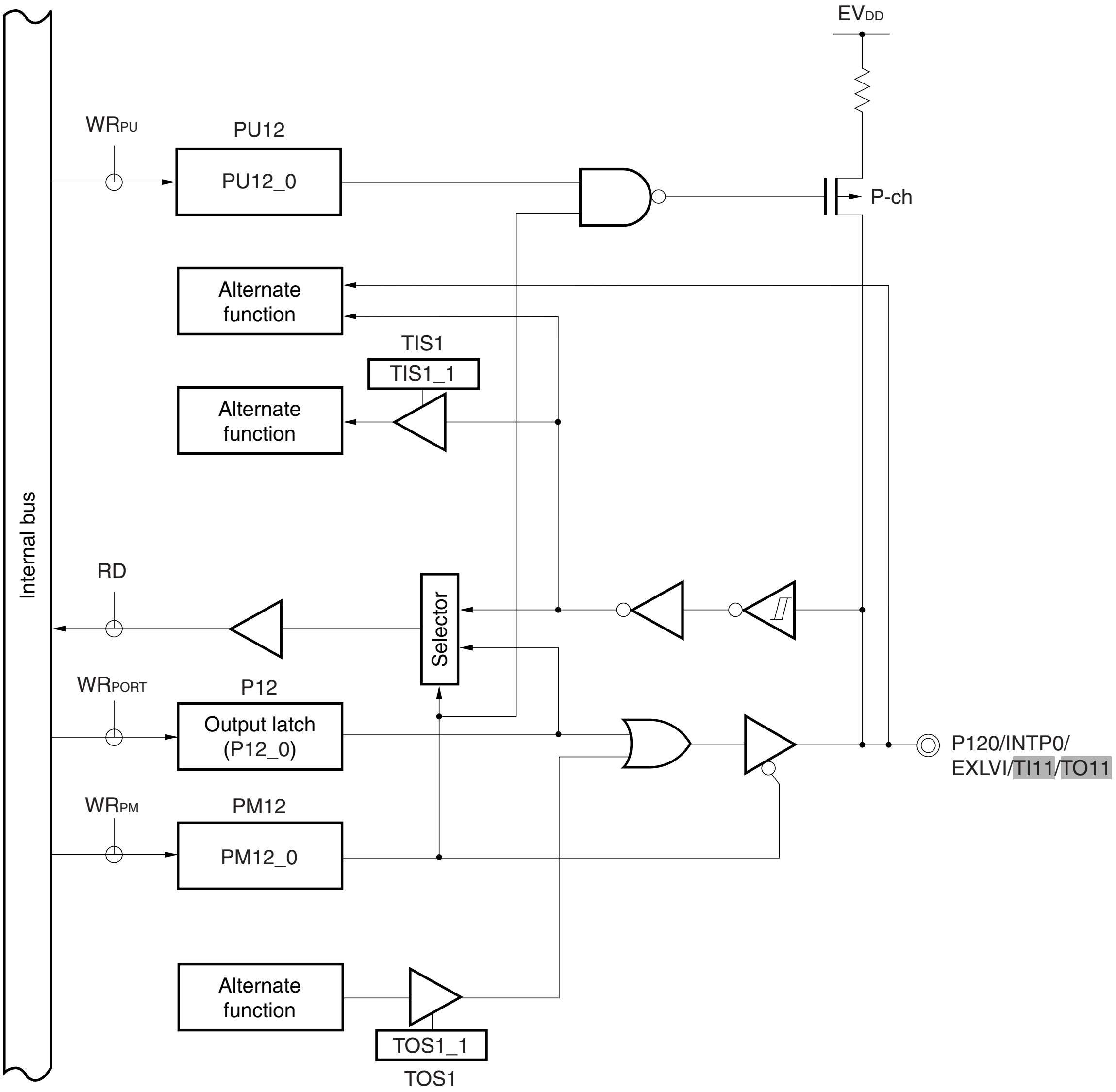
Figures 4-59 to 4-66 show block diagrams of port 12.

Cautions 1. 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.

2. To use P120/INTP0/EXLVI/TI11/TO11, P125/INTP1/ADTRG/TI03/TO03, P126/TI01/TO01, or P127/TI03/TO03 as a general-purpose port, configure bits 1 and 3 (TO01, TO03) of timer output register 0 (TO0) and bit1 (TO11) of timer output register 1 (TO1) and bits 1 and 3 (TOE0_1, TOE0_3) of timer output enable register 0 (TOE0) and bit 1 (TOE1_1) of timer output enable register 1 (TOE1) to “0”, which is the same as their default status setting.

3. The shaded pins are provided at two ports. Select either port by using the corresponding register.

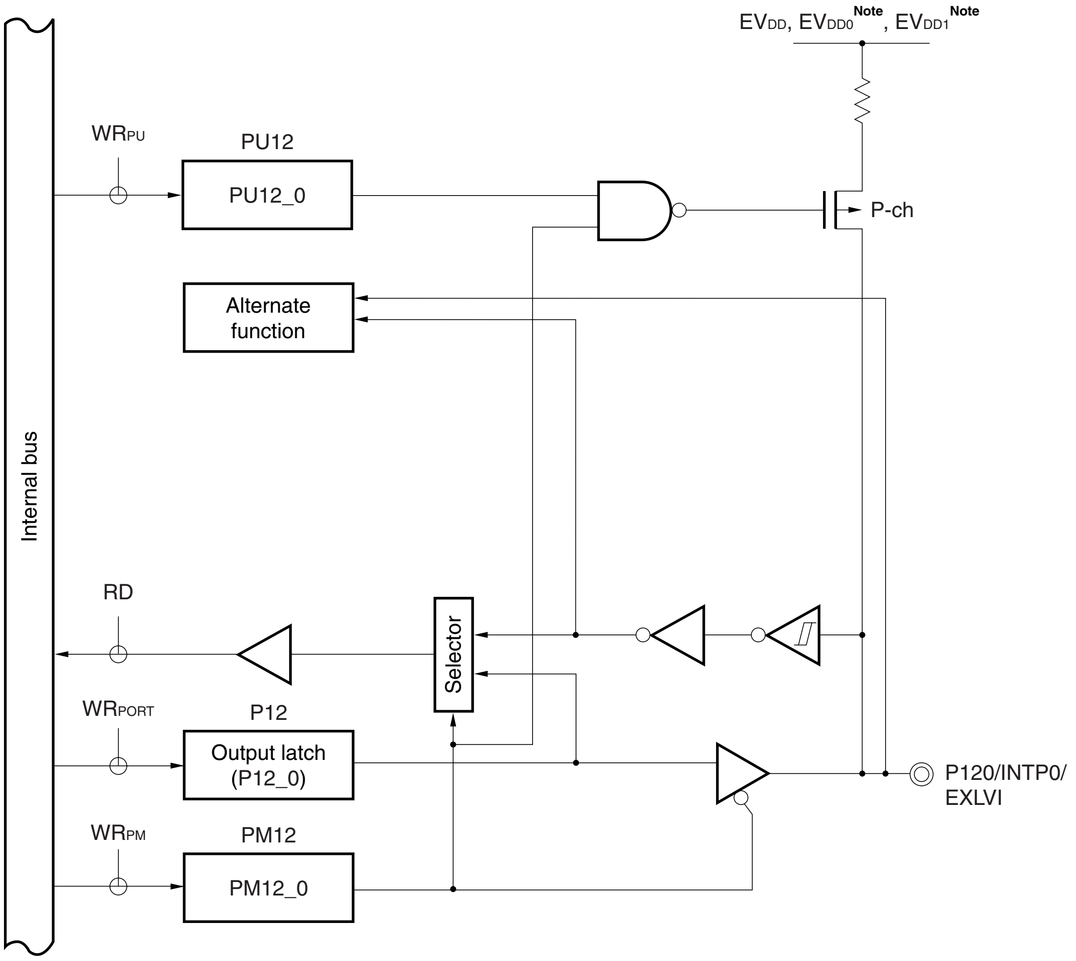
Figure 4-59. Block Diagram of P120 (78K0R/HC3, 78K0R/HE3)



Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P12:	Port register 12
PU12:	Pull-up resistor option register 12
PM12:	Port mode register 12
RD:	Read signal
TIS1:	Timer input select register 1
TOS1:	Timer output select register 1
WR××:	Write signal

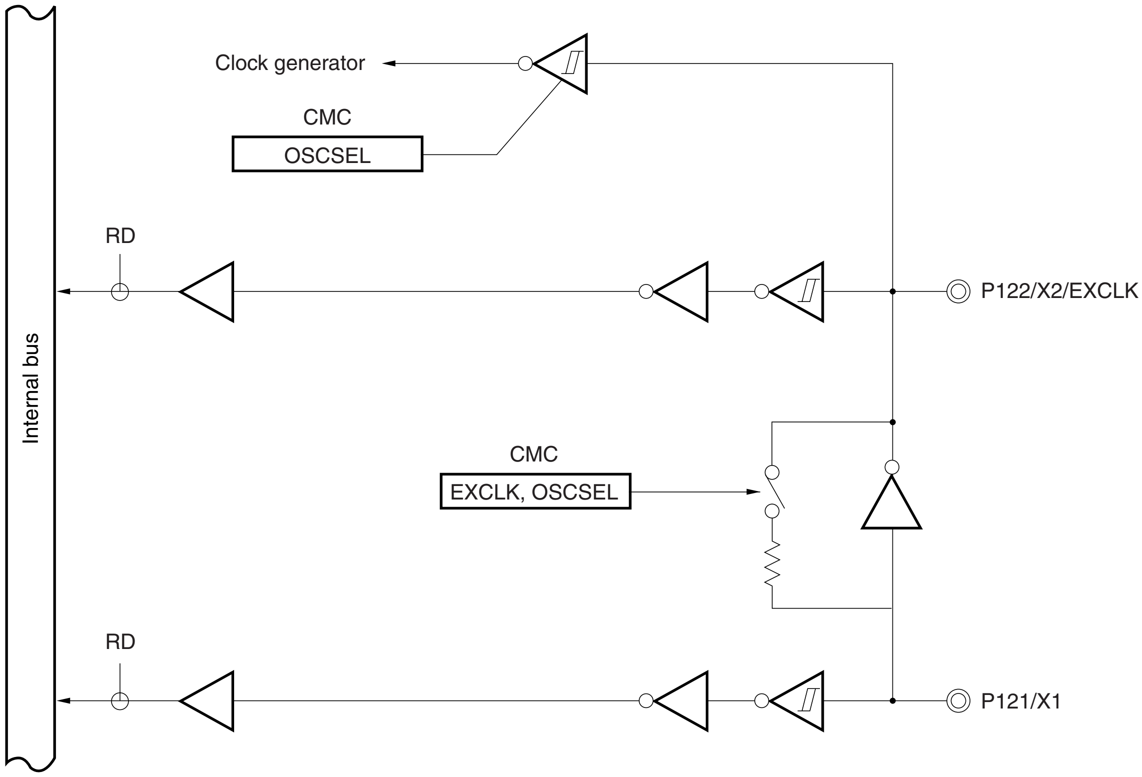
Figure 4-60. Block Diagram of P120 (78K0R/HF3, 78K0R/HG3)



Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

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

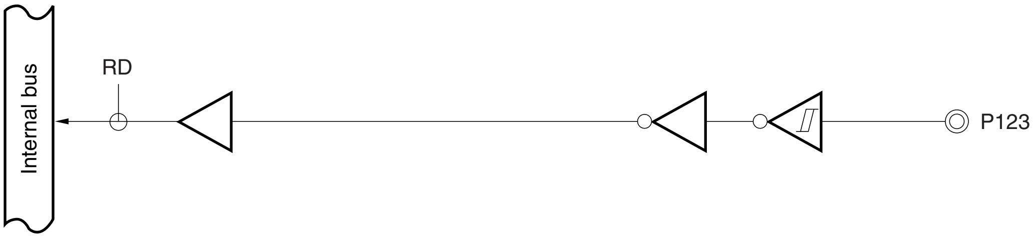
Figure 4-61. Block Diagram of P121 and P122



CMC: Clock operation mode control register

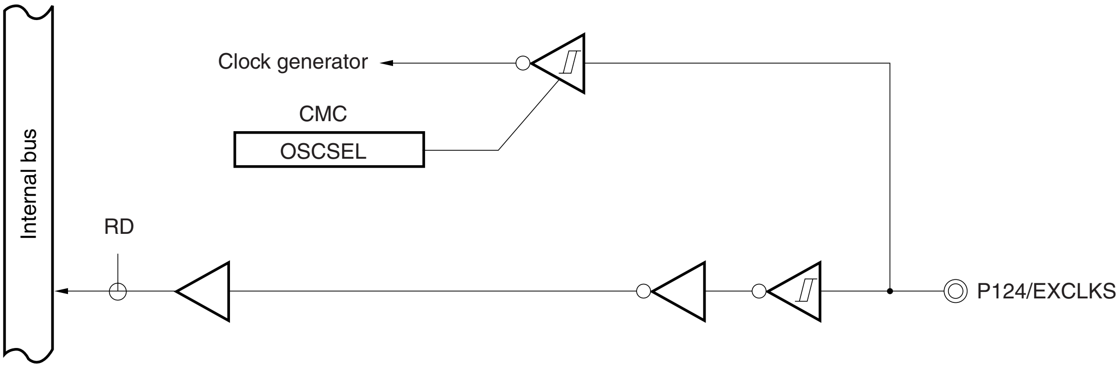
RD: Read signal

Figure 4-62. Block Diagram of P123



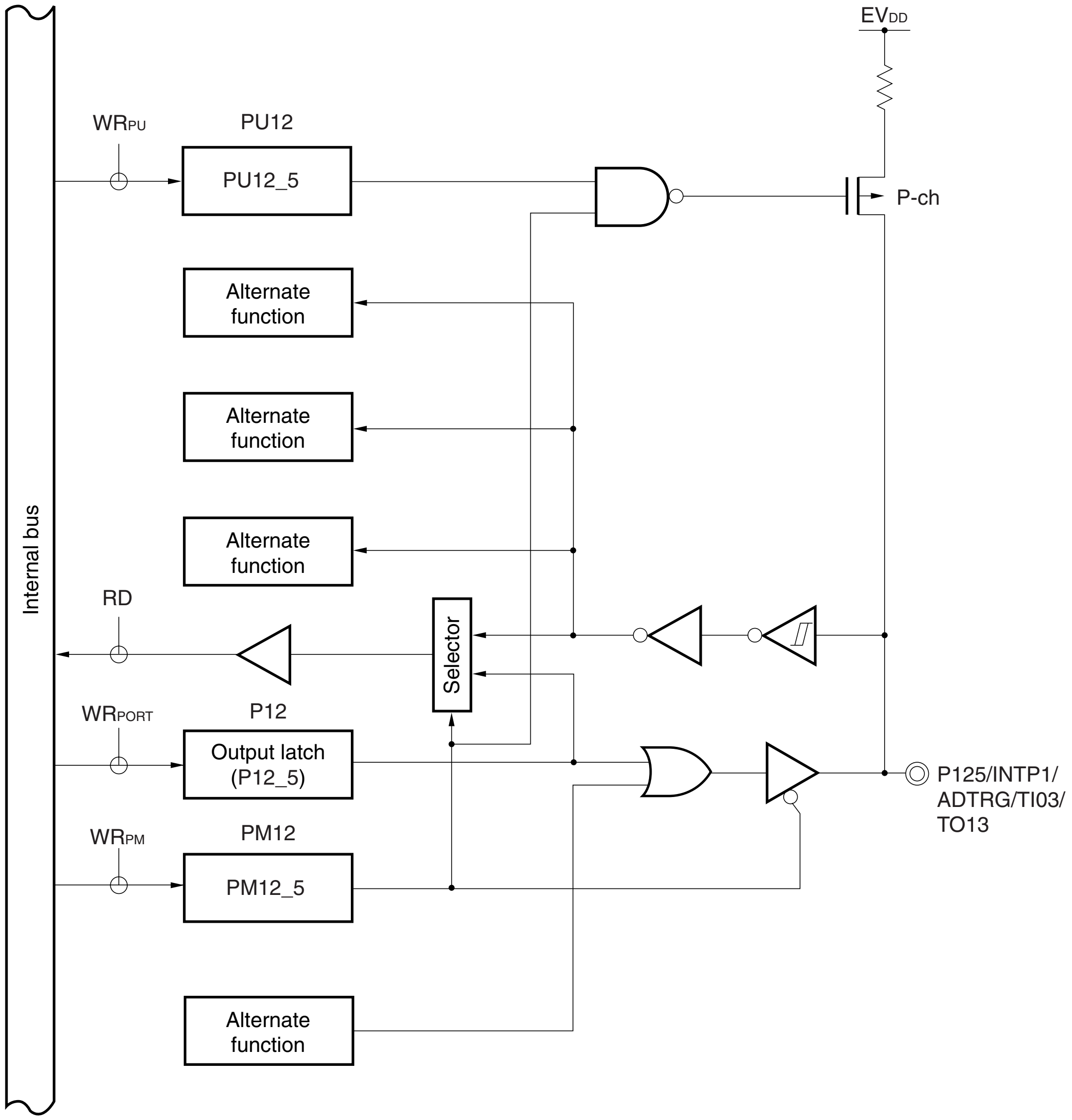
RD: Read signal

Figure 4-63. Block Diagram of P124



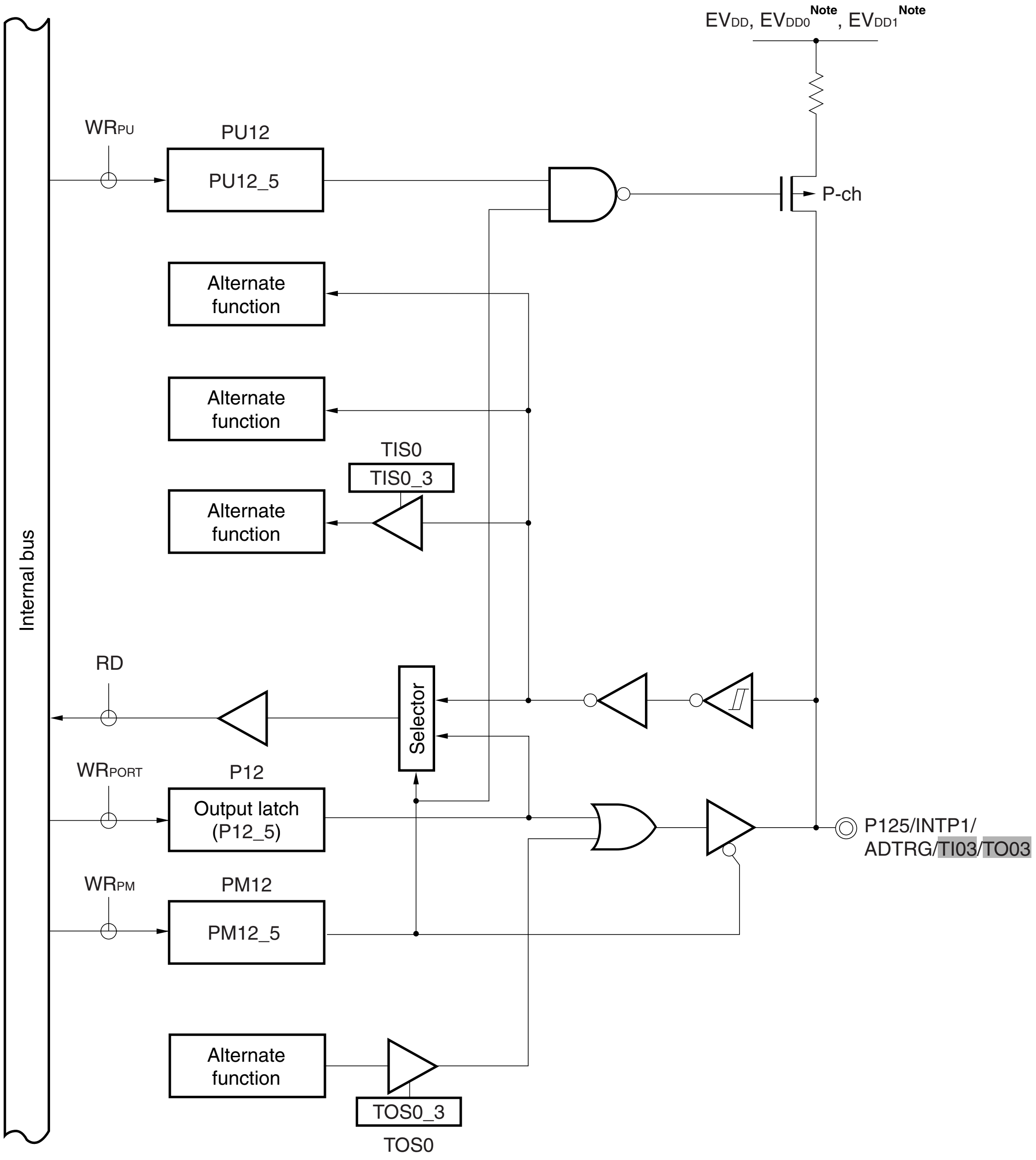
CMC: Clock operation mode control register
RD: Read signal

<R> **Figure 4-64. Block Diagram of P125 (78K0R/HC3, 78K0R/HE3, 78K0R/HF3)**



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

<R> **Figure 4-65. Block Diagram of P125 (78K0R/HG3)**

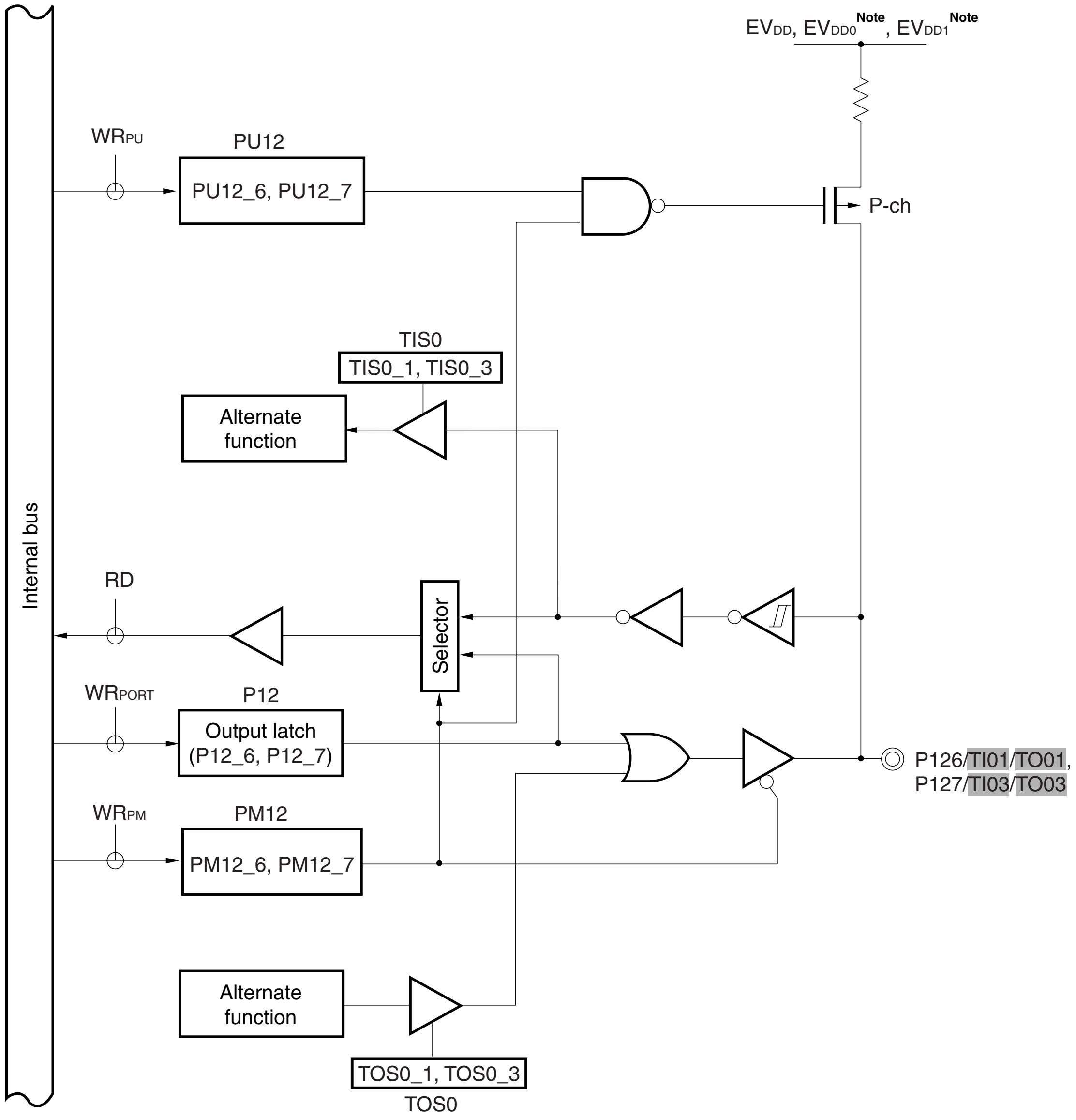


Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P12:	Port register 12
PU12:	Pull-up resistor option register 12
PM12:	Port mode register 12
RD:	Read signal
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

Figure 4-66. Block Diagram of P126 and P127



Note EV_{DD0}, EV_{DD1} are 78K0R/HG3 only

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

P12:	Port register 12
PU12:	Pull-up resistor option register 12
PM12:	Port mode register 12
RD:	Read signal
TIS0:	Timer input select register 0
TOS0:	Timer output select register 0
WR××:	Write signal

4.2.12 Port 13

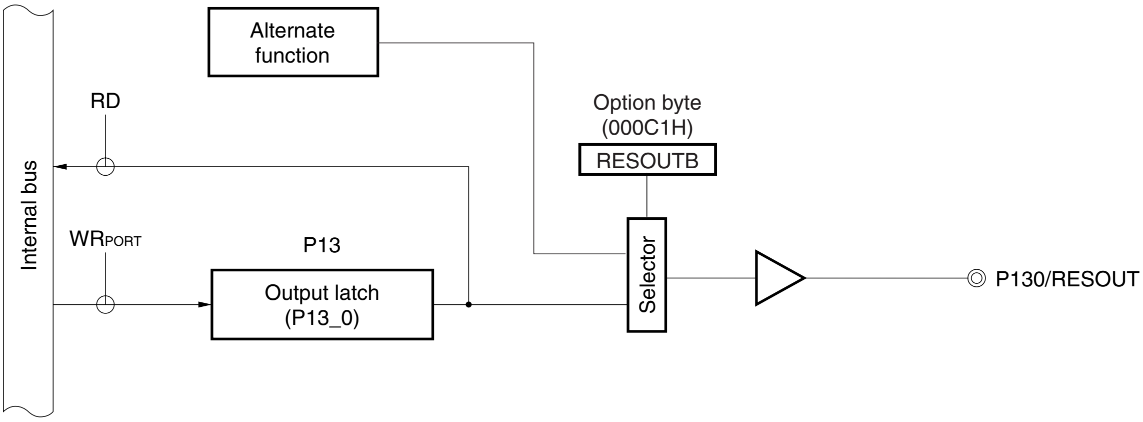
	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36- to 40	yy = 41 to 45	yy = 46 to 50
P130/RESOUT	√	√	√	√

Remarks 1. When the device is reset, P130 outputs a low level.

2. √: Mounted, −: Not mounted

P130 is a 1-bit output-only port with an output latch.
Reset signal generation sets port 13 to output mode.
This port can also be used for reset output.
Figure 4-67 shows block diagrams of port 13.

Figure 4-67. Block Diagram of P130



P13:	Port register 13
RD:	Read signal
WR××:	Write signal

4.2.13 Port 14

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P140/PCL	√	√	√	√

Remarks 1. When the device is reset, P140 outputs a low level.

2. √: Mounted, −: Not mounted

Port 14 is an I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140 pin 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 14 (PU14).

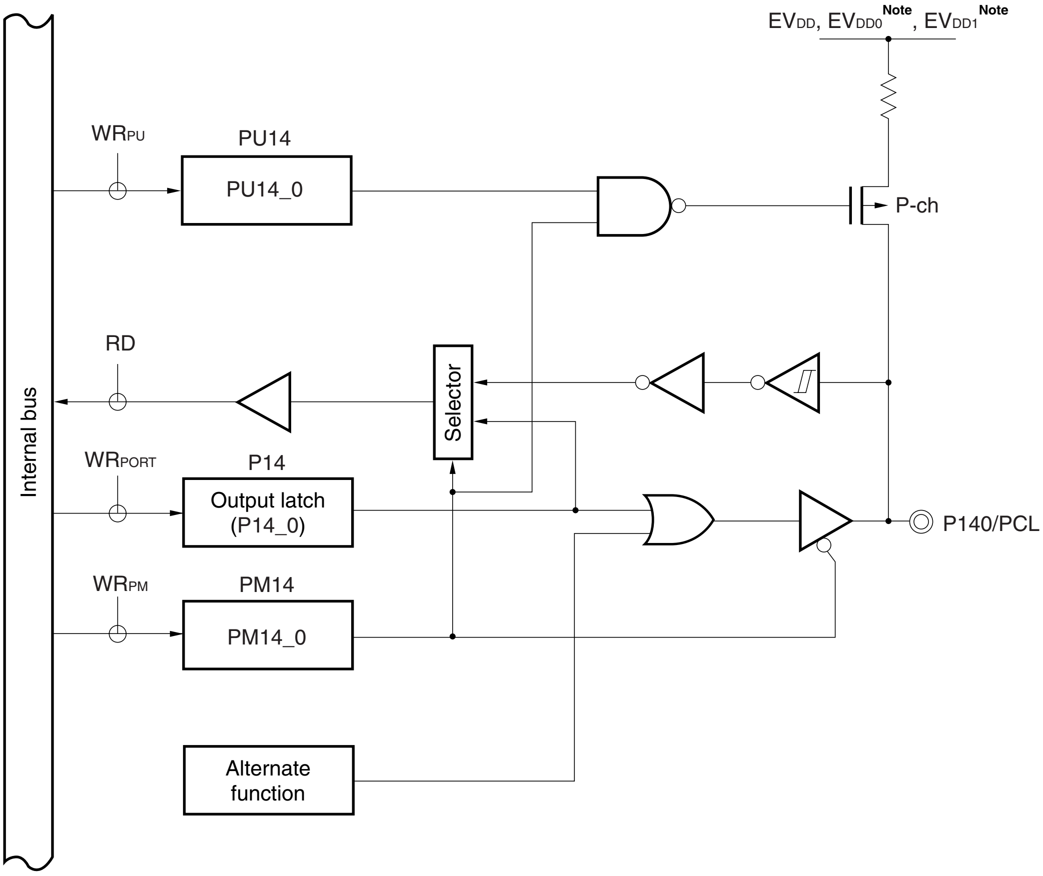
This port can also be used for clock output.

Reset signal generation sets port 14 to output mode.

Figure 4-68 shows block diagrams of port 14.

Caution To use P140/PCL as a general-purpose port, set bit 7 of clock output select register (CKS) to “0”, which is the same as their default status setting.

Figure 4-68. Block Diagram of P140



Note EV$_{DD0}$, EV$_{DD1}$ are 78K0R/HG3 only

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

4.2.14 Port 15

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
P150	–	–	–	√
P151/SO11	–	–	–	√
P152/SI11	–	–	–	√
P153/$\overline{\text{SCK11}}$	–	–	–	√
P154/TI24/TO24	–	–	–	√
P155/TI25/TO25	–	–	–	√
P156/TI26/TO26	–	–	–	√
P157/TI27/TO27	–	–	–	√

Remark √: Mounted, –: Not mounted

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). When the P150 to P157 pins 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 15 (PU15).

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

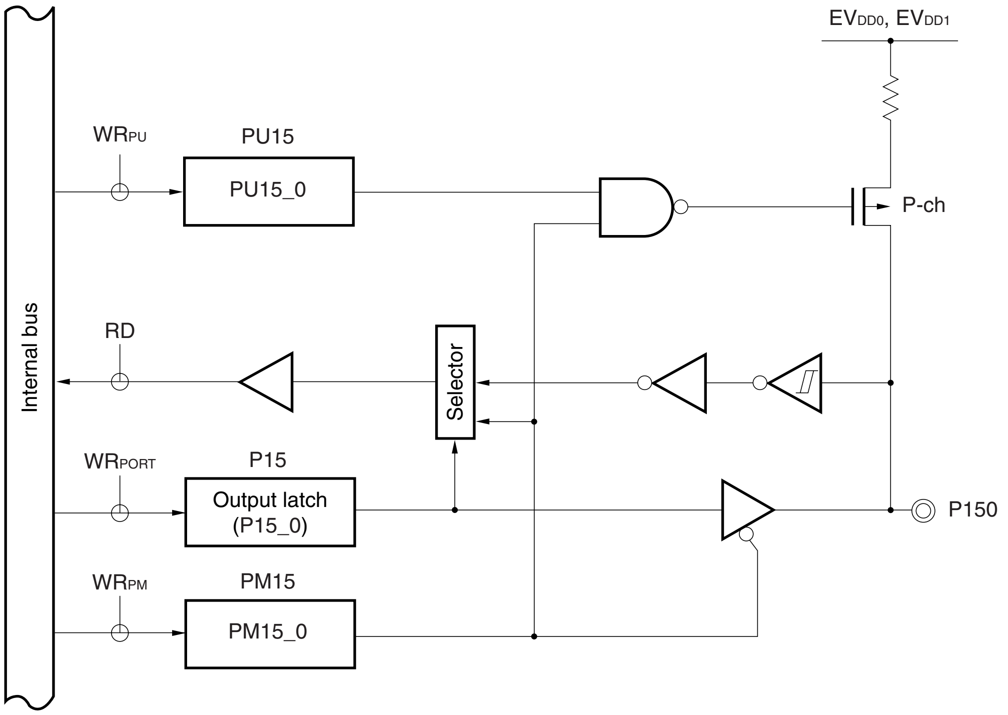
Reset signal generation sets port 15 to input mode.

Figures 4-69 to 4-73 show a block diagram of port 15.

Cautions 1. To use P151/SO11, P152/SI11, or P153/$\overline{\text{SCK11}}$ as a general-purpose port, note the serial array unit setting. For details, see Table 11-12. Relationship Between Register Settings and Pins (Channel 1 of Unit 1: CSI11, IIC11).

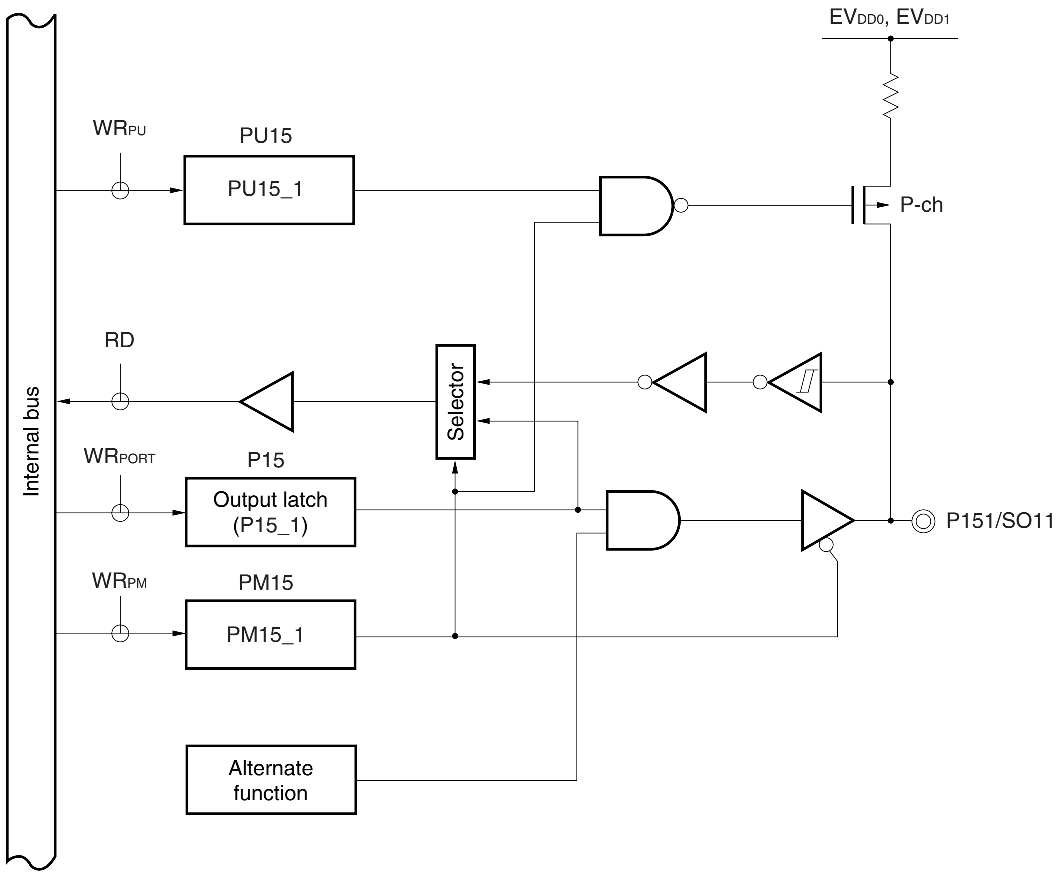
2. To use P154/TI24/TO24, P155/TI25/TO25, P156/TI26/TO26, or P157/TI27/TO27 as a general-purpose port, configure bits 4 to 7 (TO24 to TO27) of timer output register 2 (TO2) and bits 4 to 7 (TOE2_4 to TOE2_7) of timer output enable register 2 (TOE2) to "0", which is the same as their default status setting.

Figure 4-69. Block Diagram of P150



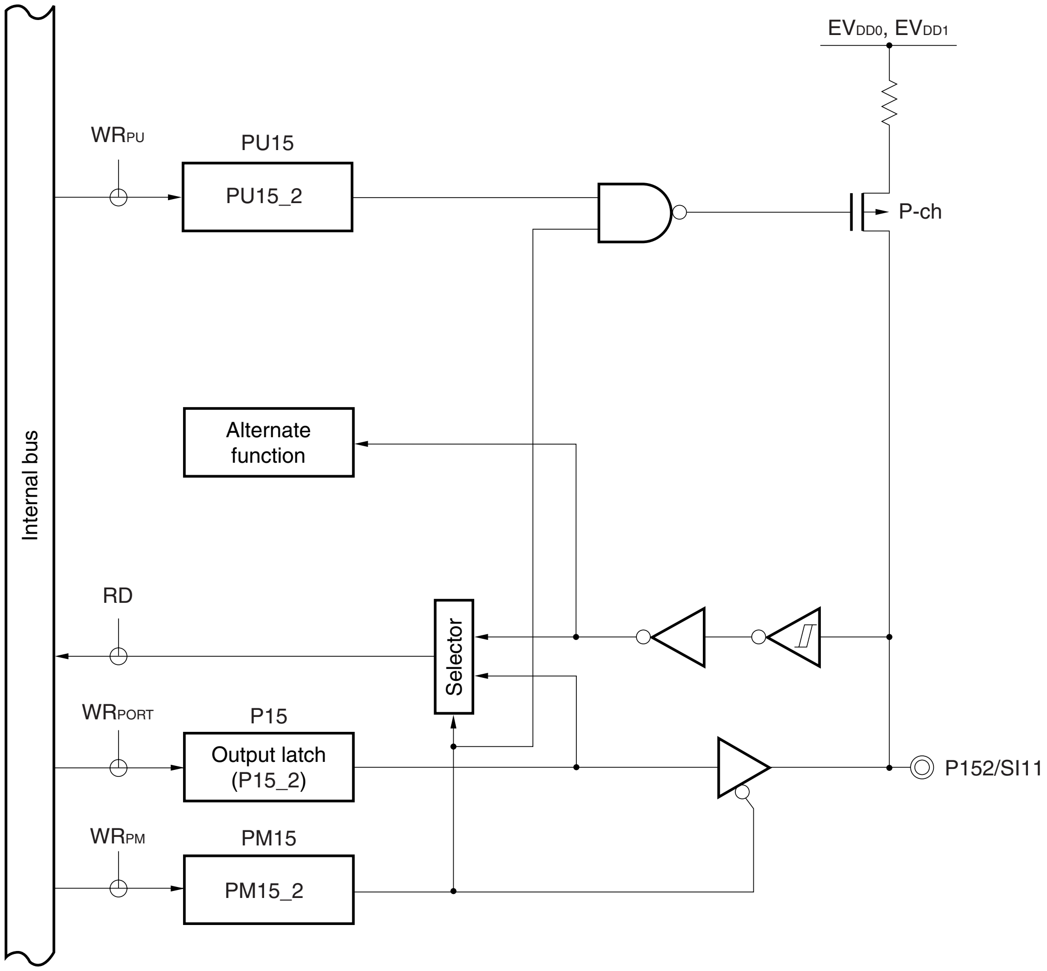
P15:	Port register 15
PU15:	Pull-up resistor option register 15
PM15:	Port mode register 15
RD:	Read signal
WR××:	Write signal

Figure 4-70. Block Diagram of P151



P15:	Port register 15
PU15:	Pull-up resistor option register 15
PM15:	Port mode register 15
RD:	Read signal
WR××:	Write signal

Figure 4-71. Block Diagram of P152



P15: Port register 15

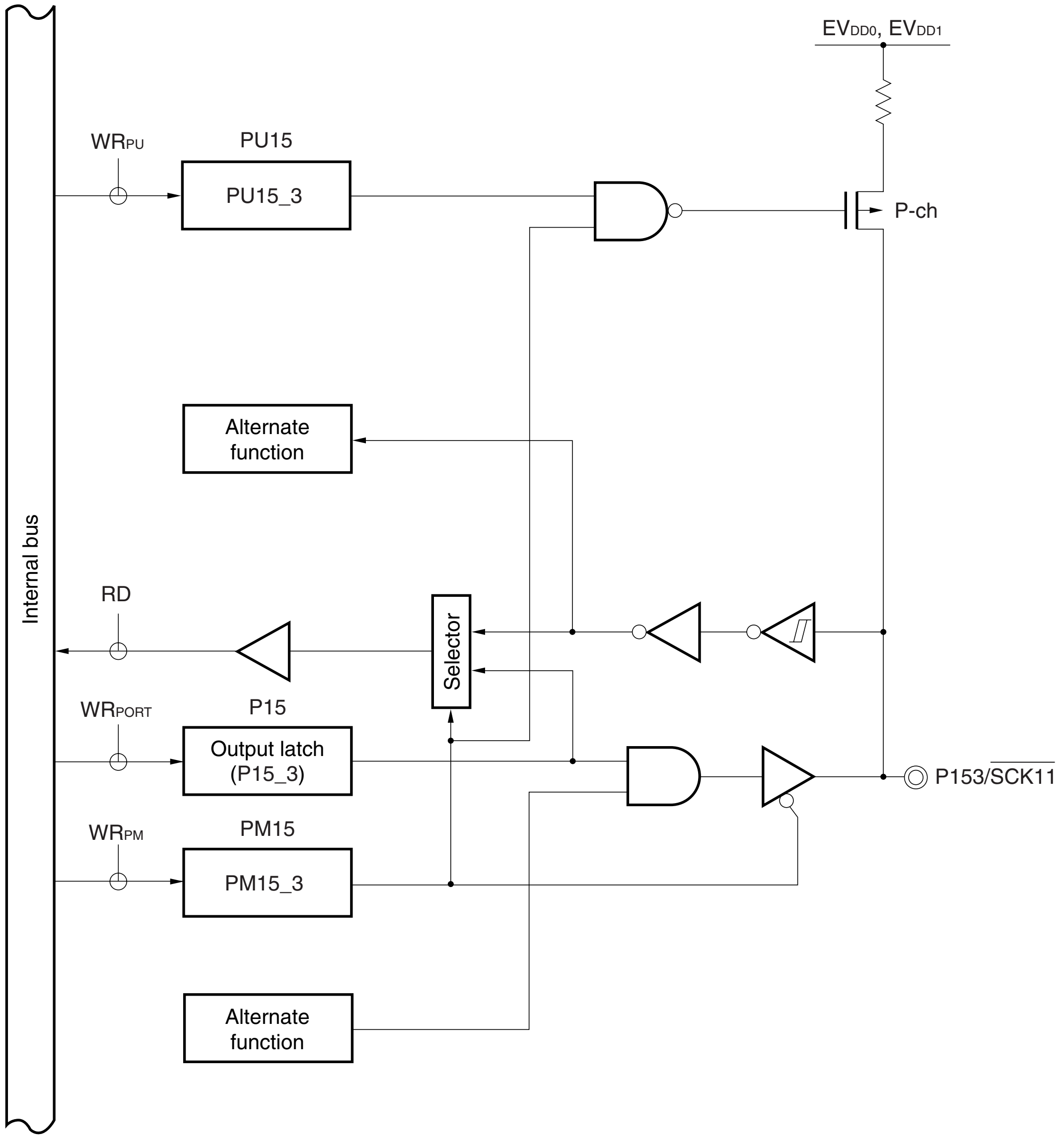
PU15: Pull-up resistor option register 15

PM15: Port mode register 15

RD: Read signal

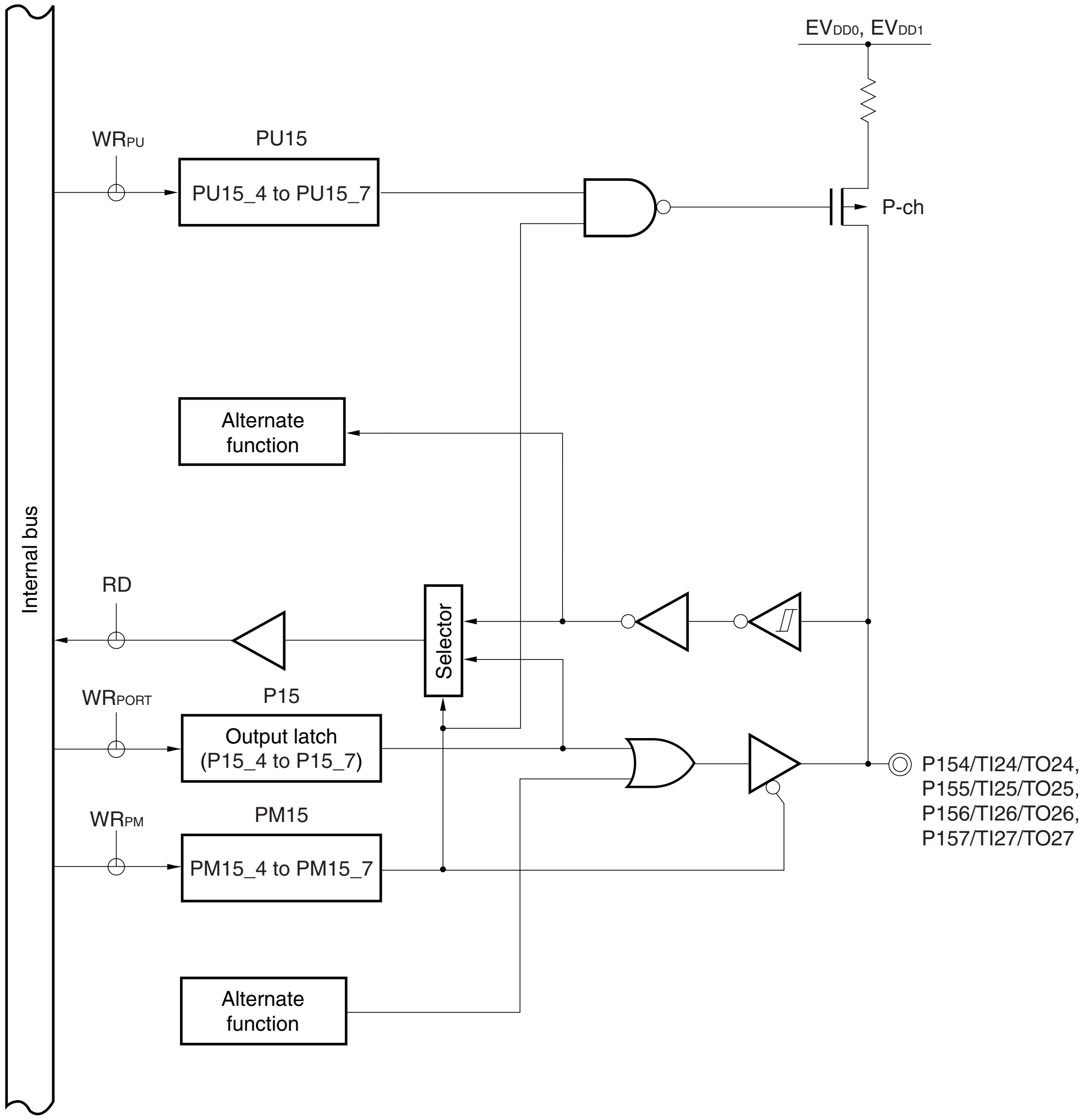
WR××: Write signal

Figure 4-72. Block Diagram of P153



P15:	Port register 15
PU15:	Pull-up resistor option register 15
PM15:	Port mode register 15
RD:	Read signal
WR××:	Write signal

Figure 4-73. Block Diagram of P154 to P157



P15:	Port register 15
PU15:	Pull-up resistor option register 15
PM15:	Port mode register 15
RD:	Read signal
WR××:	Write signal

4.3 Registers Controlling Port Function

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

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port output slew rate select register (PSRSEL)
- A/D port configuration register (ADPC)

<R> **(1) Port mode registers (PMxx)**

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

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

Reset signal generation sets these registers to FFH (PM14 is set to FEH).

When port pins are used as alternate-function pins, set the port mode register by referencing **4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function**.

Figure 4-74. Format of Port Mode Register (78K0R/HC3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	1	1	PM0_0	FFF20H	FFH	R/W
PM1	PM1_7	PM1_6	PM1_5	PM1_4	PM1_3	PM1_2	PM1_1	PM1_0	FFF21H	FFH	R/W
PM3	1	1	1	1	1	PM3_2	PM3_1	PM3_0	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	PM4_1	PM4_0	FFF24H	FFH	R/W
PM6	1	1	1	1	PM6_3	PM6_2	PM6_1	PM6_0	FFF26H	FFH	R/W
PM7	1	1	1	1	PM7_3	PM7_2	PM7_1	PM7_0	FFF27H	FFH	R/W
PM8	PM8_7	PM8_6	PM8_5	PM8_4	PM8_3	PM8_2	PM8_1	PM8_0	FFF28H	FFH	R/W
PM9	1	1	1	1	1	PM9_2	PM9_1	PM9_0	FFF29H	FFH	R/W
PM12	1	1	PM12_5	1	1	1	1	PM12_0	FFF2CH	FFH	R/W
PM14	1	1	1	1	1	1	1	PM14_0	FFF2EH	FEH	R/W

PMm_n	Pmn pin I/O mode selection (m = 0, 1, 3, 4, 6 to 9, 12, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 1 to 7 of PM0, bits 3 to 7 of PM3, bits 2 to 7 of PM4, bits 4 to 7 of PM6, bits 4 to 7 of PM7, bits 3 to 7 of PM9, bits 1 to 4, 6, 7 of PM12, and bits 1 to 7 of PM14 to “1”.

Figure 4-75. Format of Port Mode Register (78K0R/HE3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	1	1	PM0_0	FFF20H	FFH	R/W
PM1	PM1_7	PM1_6	PM1_5	PM1_4	PM1_3	PM1_2	PM1_1	PM1_0	FFF21H	FFH	R/W
PM3	1	1	1	1	1	PM3_2	PM3_1	PM3_0	FFF23H	FFH	R/W
PM4	1	1	1	1	PM4_3	PM4_2	PM4_1	PM4_0	FFF24H	FFH	R/W
PM5	1	1	1	1	PM5_3	PM5_2	PM5_1	PM5_0	FFF25H	FFH	R/W
PM6	1	1	1	1	PM6_3	PM6_2	PM6_1	PM6_0	FFF26H	FFH	R/W
PM7	PM7_7	PM7_6	PM7_5	PM7_4	PM7_3	PM7_2	PM7_1	PM7_0	FFF27H	FFH	R/W
PM8	PM8_7	PM8_6	PM8_5	PM8_4	PM8_3	PM8_2	PM8_1	PM8_0	FFF28H	FFH	R/W
PM9	1	PM9_6	PM9_5	PM9_4	PM9_3	PM9_2	PM9_1	PM9_0	FFF29H	FFH	R/W
PM12	1	1	PM12_5	1	1	1	1	PM12_0	FFF2CH	FFH	R/W
PM14	1	1	1	1	1	1	1	PM14_0	FFF2EH	FEH	R/W

PMm_n	Pmn pin I/O mode selection (m = 0, 1, 3 to 9, 12, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 1 to 7 of PM0, bits 3 to 7 of PM3, bits 4 to 7 of PM4, bits 4 to 7 of PM5, bits 4 to 7 of PM6, bit 7 of PM9, bits 1 to 4, 6, 7 of PM12, and bits 1 to 7 of PM14 to “1”.

Figure 4-76. Format of Port Mode Register (78K0R/HF3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	PM0_2	PM0_1	PM0_0	FFF20H	FFH	R/W
PM1	PM1_7	PM1_6	PM1_5	PM1_4	PM1_3	PM1_2	PM1_1	PM1_0	FFF21H	FFH	R/W
PM3	1	1	1	1	1	PM3_2	PM3_1	PM3_0	FFF23H	FFH	R/W
PM4	PM4_7	PM4_6	PM4_5	PM4_4	PM4_3	PM4_2	PM4_1	PM4_0	FFF24H	FFH	R/W
PM5	PM5_7	PM5_6	PM5_5	PM5_4	PM5_3	PM5_2	PM5_1	PM5_0	FFF25H	FFH	R/W
PM6	PM6_7	PM6_6	PM6_5	PM6_4	PM6_3	PM6_2	PM6_1	PM6_0	FFF26H	FFH	R/W
PM7	PM7_7	PM7_6	PM7_5	PM7_4	PM7_3	PM7_2	PM7_1	PM7_0	FFF27H	FFH	R/W
PM8	PM8_7	PM8_6	PM8_5	PM8_4	PM8_3	PM8_2	PM8_1	PM8_0	FFF28H	FFH	R/W
PM9	PM9_7	PM9_6	PM9_5	PM9_4	PM9_3	PM9_2	PM9_1	PM9_0	FFF29H	FFH	R/W
PM12	1	PM12_6	PM12_5	1	1	1	1	PM12_0	FFF2CH	FFH	R/W
PM14	1	1	1	1	1	1	1	PM14_0	FFF2EH	FEH	R/W

PMm_n	Pmn pin I/O mode selection (m = 0, 1, 3 to 9, 12, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 3 to 7 of PM0, and bits 3 to 7 of PM3, and bits 1 to 4, 7 of PM12, and bits 1 to 7 of PM14 to "1".

Figure 4-77. Format of Port Mode Register (78K0R/HG3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	PM0_3	PM0_2	PM0_1	PM0_0	FFF20H	FFH	R/W
PM1	PM1_7	PM1_6	PM1_5	PM1_4	PM1_3	PM1_2	PM1_1	PM1_0	FFF21H	FFH	R/W
PM3	1	1	1	1	1	PM3_2	PM3_1	PM3_0	FFF23H	FFH	R/W
PM4	PM4_7	PM4_6	PM4_5	PM4_4	PM4_3	PM4_2	PM4_1	PM4_0	FFF24H	FFH	R/W
PM5	PM5_7	PM5_6	PM5_5	PM5_4	PM5_3	PM5_2	PM5_1	PM5_0	FFF25H	FFH	R/W
PM6	PM6_7	PM6_6	PM6_5	PM6_4	PM6_3	PM6_2	PM6_1	PM6_0	FFF26H	FFH	R/W
PM7	PM7_7	PM7_6	PM7_5	PM7_4	PM7_3	PM7_2	PM7_1	PM7_0	FFF27H	FFH	R/W
PM8	PM8_7	PM8_6	PM8_5	PM8_4	PM8_3	PM8_2	PM8_1	PM8_0	FFF28H	FFH	R/W
PM9	PM9_7	PM9_6	PM9_5	PM9_4	PM9_3	PM9_2	PM9_1	PM9_0	FFF29H	FFH	R/W
PM10	PM10_7	PM10_6	PM10_5	PM10_4	PM10_3	PM10_2	PM10_1	PM10_0	FFF2AH	FFH	R/W
PM12	PM12_7	PM12_6	PM12_5	1	1	1	1	PM12_0	FFF2CH	FFH	R/W
PM14	1	1	1	1	1	1	1	PM14_0	FFF2EH	FEH	R/W
PM15	PM15_7	PM15_6	PM15_5	PM15_4	PM15_3	PM15_2	PM15_1	PM15_0	FFF2FH	FFH	R/W

PMm_n	Pmn pin I/O mode selection (m = 0, 1, 3 to 10, 12, 14, 15; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 4 to 7 of PM0, and bits 3 to 7 of PM3, and bits 1 to 4 of PM12, and bits 1 to 7 of PM14 to “1”.

<R> **(2) Port registers (Pxx)**

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].

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

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 P8 to P10 are set to function as an analog input for a A/D converter.

Figure 4-78. Format of Port Register (78K0R/HC3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	0	0	P0_0	FFF00H	00H (output latch)	R/W
P1	P1_7	P1_6	P1_5	P1_4	P1_3	P1_2	P1_1	P1_0	FFF01H	00H (output latch)	R/W
P3	0	0	0	0	0	P3_2	P3_1	P3_0	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	0	0	P4_1	P4_0	FFF04H	00H (output latch)	R/W
P6	0	0	0	0	P6_3	P6_2	P6_1	P6_0	FFF06H	00H (output latch)	R/W
P7	0	0	0	0	P7_3	P7_2	P7_1	P7_0	FFF07H	00H (output latch)	R/W
P8	P8_7	P8_6	P8_5	P8_4	P8_3	P8_2	P8_1	P8_0	FFF08H	00H (output latch)	R/W
P9	0	0	0	0	0	P9_2	P9_1	P9_0	FFF09H	00H (output latch)	R/W
P12	0	0	P12_5	P12_4	P12_3	P12_2	P12_1	P12_0	FFF0CH	00H (output latch)	R/W[Note]
P13	0	0	0	0	0	0	0	P13_0	FFF0DH	00H (output latch)	R/W
P14	0	0	0	0	0	0	0	P14_0	FFF0EH	00H (output latch)	R/W

Pm_n	m = 0, 1, 3 to 9, 12 to 14; 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

Note P12_1 to P12_4 are read-only.

Figure 4-79. Format of Port Register (78K0R/HE3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	0	0	P0_0	FFF00H	00H (output latch)	R/W
P1	P1_7	P1_6	P1_5	P1_4	P1_3	P1_2	P1_1	P1_0	FFF01H	00H (output latch)	R/W
P3	0	0	0	0	0	P3_2	P3_1	P3_0	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	P4_3	P4_2	P4_1	P4_0	FFF04H	00H (output latch)	R/W
P5	0	0	0	0	P5_3	P5_2	P5_1	P5_0	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	P6_3	P6_2	P6_1	P6_0	FFF06H	00H (output latch)	R/W
P7	P7_7	P7_6	P7_5	P7_4	P7_3	P7_2	P7_1	P7_0	FFF07H	00H (output latch)	R/W
P8	P8_7	P8_6	P8_5	P8_4	P8_3	P8_2	P8_1	P8_0	FFF08H	00H (output latch)	R/W
P9	0	P9_6	P9_5	P9_4	P9_3	P9_2	P9_1	P9_0	FFF09H	00H (output latch)	R/W
P12	0	0	P12_5	P12_4	P12_3	P12_2	P12_1	P12_0	FFF0CH	00H (output latch)	R/W[Note]
P13	0	0	0	0	0	0	0	P13_0	FFF0DH	00H (output latch)	R/W
P14	0	0	0	0	0	0	0	P14_0	FFF0EH	00H (output latch)	R/W

Pm_n	m = 0, 1, 3 to 9, 12 to 14; 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

Note P12_1 to P12_4 are read-only.

Figure 4-80. Format of Port Register (78K0R/HF3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	P0_2	P0_1	P0_0	FFF00H	00H (output latch)	R/W
P1	P1_7	P1_6	P1_5	P1_4	P1_3	P1_2	P1_1	P1_0	FFF01H	00H (output latch)	R/W
P3	0	0	0	0	0	P3_2	P3_1	P3_0	FFF03H	00H (output latch)	R/W
P4	P4_7	P4_6	P4_5	P4_4	P4_3	P4_2	P4_1	P4_0	FFF04H	00H (output latch)	R/W
P5	P5_7	P5_6	P5_5	P5_4	P5_3	P5_2	P5_1	P5_0	FFF05H	00H (output latch)	R/W
P6	P6_7	P6_6	P6_5	P6_4	P6_3	P6_2	P6_1	P6_0	FFF06H	00H (output latch)	R/W
P7	P7_7	P7_6	P7_5	P7_4	P7_3	P7_2	P7_1	P7_0	FFF07H	00H (output latch)	R/W
P8	P8_7	P8_6	P8_5	P8_4	P8_3	P8_2	P8_1	P8_0	FFF08H	00H (output latch)	R/W
P9	P9_7	P9_6	P9_5	P9_4	P9_3	P9_2	P9_1	P9_0	FFF09H	00H (output latch)	R/W
P12	0	P12_6	P12_5	P12_4	P12_3	P12_2	P12_1	P12_0	FFF0CH	00H (output latch)	R/W[Note]
P13	0	0	0	0	0	0	0	P13_0	FFF0DH	00H (output latch)	R/W
P14	0	0	0	0	0	0	0	P14_0	FFF0EH	00H (output latch)	R/W

Pmn	m = 0, 1, 3 to 9, 12 to 14; 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

Note P12_1 to P12_4 are read-only.

Figure 4-81. Format of Port Register (78K0R/HG3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	P0_3	P0_2	P0_1	P0_0	FFF00H	00H (output latch)	R/W
P1	P1_7	P1_6	P1_5	P1_4	P1_3	P1_2	P1_1	P1_0	FFF01H	00H (output latch)	R/W
P3	0	0	0	0	0	P3_2	P3_1	P3_0	FFF03H	00H (output latch)	R/W
P4	P4_7	P4_6	P4_5	P4_4	P4_3	P4_2	P4_1	P4_0	FFF04H	00H (output latch)	R/W
P5	P5_7	P5_6	P5_5	P5_4	P5_3	P5_2	P5_1	P5_0	FFF05H	00H (output latch)	R/W
P6	P6_7	P6_6	P6_5	P6_4	P6_3	P6_2	P6_1	P6_0	FFF06H	00H (output latch)	R/W
P7	P7_7	P7_6	P7_5	P7_4	P7_3	P7_2	P7_1	P7_0	FFF07H	00H (output latch)	R/W
P8	P8_7	P8_6	P8_5	P8_4	P8_3	P8_2	P8_1	P8_0	FFF08H	00H (output latch)	R/W
P9	P9_7	P9_6	P9_5	P9_4	P9_3	P9_2	P9_1	P9_0	FFF09H	00H (output latch)	R/W
P10	P10_7	P10_6	P10_5	P10_4	P10_3	P10_2	P10_1	P10_0	FFF0AH	00H (output latch)	R/W
P12	P12_7	P12_6	P12_5	P12_4	P12_3	P12_2	P12_1	P12_0	FFF0CH	00H (output latch)	R/W[Note]
P13	0	0	0	0	0	0	0	P13_0	FFF0DH	00H (output latch)	R/W
P14	0	0	0	0	0	0	0	P14_0	FFF0EH	00H (output latch)	R/W
P15	P15_7	P15_6	P15_5	P15_4	P15_3	P15_2	P15_1	P15_0	FFF0FH	00H (output latch)	R/W

Pm_n	m = 0, 1, 3 to 10, 12 to 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

Note P12_1 to P12_4 are read-only.

<R> (3) Pull-up resistor option registers (PUxx)

These registers specify whether the on-chip pull-up resistors are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to 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.

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

Reset signal generation clears these registers to 00H.

Figure 4-82. Format of Pull-up Resistor Option Register (78K0R/HC3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	0	0	PU0_0	F0030H	00H	R/W
PU1	PU1_7	PU1_6	PU1_5	PU1_4	PU1_3	PU1_2	PU1_1	PU1_0	F0031H	00H	R/W
PU3	0	0	0	0	0	PU3_2	PU3_1	PU3_0	F0033H	00H	R/W
PU4	0	0	0	0	0	0	PU4_1	PU4_0	F0034H	00H	R/W
PU6	0	0	0	0	PU6_3	PU6_2	PU6_1	PU6_0	F0036H	00H	R/W
PU7	0	0	0	0	PU7_3	PU7_2	PU7_1	PU7_0	F0037H	00H	R/W
PU12	0	0	PU12_5	0	0	0	0	PU12_0	F003CH	00H	R/W
PU14	0	0	0	0	0	0	0	PU14_0	F003EH	00H	R/W

PUm_n	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3, 4, 6, 7, 12, 14; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Figure 4-83. Format of Pull-up Resistor Option Register (78K0R/HE3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	0	0	PU0_0	F0030H	00H	R/W
PU1	PU1_7	PU1_6	PU1_5	PU1_4	PU1_3	PU1_2	PU1_1	PU1_0	F0031H	00H	R/W
PU3	0	0	0	0	0	PU3_2	PU3_1	PU3_0	F0033H	00H	R/W
PU4	0	0	0	0	PU4_3	PU4_2	PU4_1	PU4_0	F0034H	00H	R/W
PU5	0	0	0	0	PU5_3	PU5_2	PU5_1	PU5_0	F0035H	00H	R/W
PU6	0	0	0	0	PU6_3	PU6_2	PU6_1	PU6_0	F0036H	00H	R/W
PU7	PU7_7	PU7_6	PU7_5	PU7_4	PU7_3	PU7_2	PU7_1	PU7_0	F0037H	00H	R/W
PU12	0	0	PU12_5	0	0	0	0	PU12_0	F003CH	00H	R/W
PU14	0	0	0	0	0	0	0	PU14_0	F003EH	00H	R/W

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

Figure 4-84. Format of Pull-up Resistor Option Register (78K0R/HF3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	PU0_2	PU0_1	PU0_0	F0030H	00H	R/W
PU1	PU1_7	PU1_6	PU1_5	PU1_4	PU1_3	PU1_2	PU1_1	PU1_0	F0031H	00H	R/W
PU3	0	0	0	0	0	PU3_2	PU3_1	PU3_0	F0033H	00H	R/W
PU4	PU4_7	PU4_6	PU4_5	PU4_4	PU4_3	PU4_2	PU4_1	PU4_0	F0034H	00H	R/W
PU5	PU5_7	PU5_6	PU5_5	PU5_4	PU5_3	PU5_2	PU5_1	PU5_0	F0035H	00H	R/W
PU6	PU6_7	PU6_6	PU6_5	PU6_4	PU6_3	PU6_2	PU6_1	PU6_0	F0036H	00H	R/W
PU7	PU7_7	PU7_6	PU7_5	PU7_4	PU7_3	PU7_2	PU7_1	PU7_0	F0037H	00H	R/W
PU12	0	PU12_6	PU12_5	0	0	0	0	PU12_0	F003CH	00H	R/W
PU14	0	0	0	0	0	0	0	PU14_0	F003EH	00H	R/W

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

Figure 4-85. Format of Pull-up Resistor Option Register (78K0R/HG3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	PU0_3	PU0_2	PU0_1	PU0_0	F0030H	00H	R/W
PU1	PU1_7	PU1_6	PU1_5	PU1_4	PU1_3	PU1_2	PU1_1	PU1_0	F0031H	00H	R/W
PU3	0	0	0	0	0	PU3_2	PU3_1	PU3_0	F0033H	00H	R/W
PU4	PU4_7	PU4_6	PU4_5	PU4_4	PU4_3	PU4_2	PU4_1	PU4_0	F0034H	00H	R/W
PU5	PU5_7	PU5_6	PU5_5	PU5_4	PU5_3	PU5_2	PU5_1	PU5_0	F0035H	00H	R/W
PU6	PU6_7	PU6_6	PU6_5	PU6_4	PU6_3	PU6_2	PU6_1	PU6_0	F0036H	00H	R/W
PU7	PU7_7	PU7_6	PU7_5	PU7_4	PU7_3	PU7_2	PU7_1	PU7_0	F0037H	00H	R/W
PU12	PU12_7	PU12_6	PU12_5	0	0	0	0	PU12_0	F003CH	00H	R/W
PU14	0	0	0	0	0	0	0	PU14_0	F003EH	00H	R/W
PU15	PU15_7	PU15_6	PU15_5	PU15_4	PU15_3	PU15_2	PU15_1	PU15_0	F003FH	00H	R/W

PUm_n	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 7, 12, 14, 15; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

<R> **(4) Port input mode registers (PIMxx)**

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

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

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

Reset signal generation clears these registers to 00H.

Figure 4-86. Format of Port Input Mode Register (78K0R/HC3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM6	0	0	0	0	PIM6_3	0	PIM6_1	PIM6_0	F0046H	00H	R/W
PIM7	0	0	0	0	PIM7_3	0	0	0	F0047H	00H	R/W

PIMm_n	Pmn pin input buffer selection (m = 6, 7; n = 0, 1, 3)
0	Normal input buffer
1	TTL input buffer

Figure 4-87. Format of Port Input Mode Register (78K0R/HE3, 78K0R/HF3, 78K0R/HG3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM6	0	0	0	0	PIM6_3	0	PIM6_1	PIM6_0	F0046H	00H	R/W
PIM7	PIM7_7	PIM7_6	PIM7_5	0	PIM7_3	0	0	0	F0047H	00H	R/W

PIMm_n	Pmn pin input buffer selection (m = 6, 7; n = 0, 1, 3, 5 to 7)
0	Normal input buffer
1	TTL input buffer

<R> (5) Port output mode registers (POM0, POM4, POM9, POM12, POM14)

These registers set the output mode of port 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 SDAxx pins during simplified I^{2}C 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 4-88. Format of Port Output Mode Register (78K0R/HC3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM7	0	0	0	0	0	POM7_2	0	0	F0057H	00H	R/W

POM7_2	P72 pin output mode selection
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

Figure 4-89. Format of Port Output Mode Register (78K0R/HE3, 78K0R/HF3, 78K0R/HG3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM4	0	0	0	0	POM4_3	POM4_2	0	0	F0054H	00H	R/W
POM7	0	POM7_6	0	POM7_4	0	POM7_2	0	0	F0057H	00H	R/W

POMm_n	Pmn pin output mode selection (m = 4, 7; n = 2 to 4, 6)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

<R> **(6) Port output slew rate select register (PSRSEL)**

This register selects the output slew rate of a port.

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

Reset signal generation clears this register to 00H.

Figure 4-90. Format of Port Output Slew Rate Select Register (PRSEL) (78K0R/HC3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PSRSEL	0	0	PSR140	0	0	PSR30	PSR12	PSR10	F006FH	00H	R/W

PSR140	P140/PCL pin output mode selection
0	Normal mode (5 ns/5 V (TYP.))
1	Slow mode (25 ns/5 V (TYP.))

PSR30	P30/INTP2/$\overline{\text{SSI00}}$/TI01/TO01 pin output mode selection
0	Normal mode (5 ns/5 V (TYP.))
1	Slow mode (25 ns/5 V (TYP.))

PSR12	P12/INTP3/TI16/SO10/TO16 pin output mode selection
0	Normal mode (5 ns/5 V (TYP.))
1	Slow mode (25 ns/5 V (TYP.))

PSR10	P10/TI00/$\overline{\text{SCK10}}$/TO00/CTxD/LTxD1 pin output mode selection
0	Normal mode (5 ns/5 V (TYP.))
1	Slow mode (25 ns/5 V (TYP.))

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Figure 4-91. Format of Port Output Slew Rate Select Register (PRSEL) (78K0R/HE3, 78K0R/HF3, 78K0R/HG3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PSRSEL	0	0	PSR140	PSR76	PSR74	PSR30	PSR12	PSR10	F006FH	00H	R/W

PSR140	P140/PCL pin output mode selection
0	Normal mode (5 ns/5 V (TYP.))
1	Slow mode (25 ns/5 V (TYP.))

PSR76	P76/KR6/$\overline{\text{SCK01}}$ pin output mode selection
0	Normal mode (5 ns/5 V (TYP.))
1	Slow mode (25 ns/5 V (TYP.))

PSR74	P74/KR4/SO01 pin output mode selection
0	Normal mode (5 ns/5 V (TYP.))
1	Slow mode (25 ns/5 V (TYP.))

PSR30	P30/INTP2/$\overline{\text{SSI00}}$/TI01/TO01 pin output mode selection
0	Normal mode (5 ns/5 V (TYP.))
1	Slow mode (25 ns/5 V (TYP.))

PSR12	P12/INTP3/TI16/SO10/TO16 pin output mode selection
0	Normal mode (5 ns/5 V (TYP.))
1	Slow mode (25 ns/5 V (TYP.))

PSR10	P10/TI00/$\overline{\text{SCK10}}$/TO00/CTxD/LTxD1 pin output mode selection
0	Normal mode (5 ns/5 V (TYP.))
1	Slow mode (25 ns/5 V (TYP.))

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

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

This register switches the ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97 and ANI16/P100 to ANI23/P107 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 00H[Note].

Note The ADPC register is not reset even if PER0.ADCEN = 0 is set.

Figure 4-92. Format of A/D Port Configuration Register (ADPC)

Address: F0017H After reset: 00H[Note] R/W

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

ADPC bit					Analog input (A)/digital I/O (D) switching																							
4	3	2	1	0	ANI23/P107	ANI22/P106	ANI21/P105	ANI20/P104	ANI19/P103	ANI18/P102	ANI17/P101	ANI16/P100	ANI15/P97	ANI14/P96	ANI13/P95	ANI12/P94	ANI11/P93	ANI10/P92	ANI09/P91	ANI08/P90	ANI07/P87	ANI06/P86	ANI05/P85	ANI04/P84	ANI03/P83	ANI02/P82	ANI01/P81	ANI00/P80
0	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
0	0	0	0	1	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A
0	0	0	1	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A
0	0	0	1	1	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A
0	0	1	0	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A
0	0	1	0	1	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A
0	0	1	1	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A
0	0	1	1	1	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A
0	1	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A
0	1	0	0	1	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A
0	1	0	1	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A
0	1	0	1	1	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A
0	1	1	0	0	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A
0	1	1	0	1	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A
0	1	1	1	0	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A
0	1	1	1	1	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	0	0	0	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	0	0	1	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	0	1	0	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	0	1	1	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	1	0	0	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	1	0	1	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	1	1	0	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	1	1	1	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	1	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Other than above					Setting prohibited																							

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

Note **The ADPC register is not reset even if PER0.ADCEN = 0 is set.**

Cautions

1. **Set a channel to be used for A/D conversion in the input mode by using port mode register 8 to 10 (PM8 to PM10).**
2. **Do not set the pin that is set by ADPC as digital I/O by ADS.**
3. **P80/ANI00 to P87/ANI07, P90/ANI08 to P97/ANI15 and P100/ANI16 to P107/ANI23 set as analog inputs in the order of P80/ANI00, ..., P87/ANI07, P90/ANI08, ..., P97/ANI15, P100/ANI16, ..., P107/ANI23 by the A/D port configuration register (ADPC). When using P80/ANI00 to P87/ANI07, P90/ANI08 to P97/ANI15 and P100/ANI16 to P107/ANI23 as analog inputs, start designing from P80/ANI00.**
4. **Be sure to first set the ADCEN bit of peripheral enable register 0 (PER0) to 1 when setting up the ADPC register. If ADCEN = 0, writing to the ADPC register is ignored and specified values are returned to the initial values.**

Remark P80/ANI00 to P87/ANI07, P90/ANI08 to P92/ANI10: 78K0R/HC3
P80/ANI00 to P87/ANI07, P90/ANI08 to P96/ANI14: 78K0R/HE3
P80/ANI00 to P87/ANI07, P90/ANI08 to P97/ANI15: 78K0R/HF3
P80/ANI00 to P87/ANI07, P90/ANI08 to P97/ANI15, P100/ANI16 to P107/ANI23: 78K0R/HG3

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 (3 V)

When parts of ports 4, 6, and 7 operate with V_{DD} = 4.0 V to 5.5 V, I/O connections with an external device that operates on 3 V power supply voltage are possible.

Regarding inputs, CMOS/TTL switching is possible on a bit-by-bit basis by port input mode registers (PIM6, PIM7).

Moreover, regarding outputs, different power potentials can be supported by switching the output buffer to the N-ch open drain (V_{DD} withstand voltage) by the port output mode registers (POM4, POM7).

(1) Setting procedure when using I/O pins of CSI00 and CSI01 functions

(a) Use as 3 V 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 CSI00: P60, P61, P63
In case of CSI01: P75 to P77

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

<4> V_{IH}/V_{IL} operates on 3 V operating voltage.

Remarks 1. V_{IL} use this operation within a range that satisfies the DC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

2.

78K0R/HC3:	n = 6
78K0R/HE3, 78K0R/HF3, 78K0R/HG3:	n = 6, 7

(b) Use as 3 V output port

<1> After reset release, the port mode changes to 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 CSI01: P74, P76

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

<4> Set the corresponding bit of the POM7 register to 1 to set the N-ch open drain output (V_{DD} withstand voltage) mode.

<5> Set the output mode by manipulating the PM7 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 IIC11 function

<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: P60, P61

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

<4> Set the corresponding bit of the PM6 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.

<5> Enable the operation of the serial array unit and set the mode to the simplified I^2C mode.

(3) Setting procedure when using I/O pins of LIN-UART and CAN functions

(a) Use as 3 V 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 P73 pin (on-chip pull-up resistor cannot be used).

<3> Set the corresponding bit of the PIM7 register to 1 to switch to the TTL input buffer.

<4> V_{IH}/V_{IL} operates on 3 V operating voltage.

Remark V_{IL} use this operation within a range that satisfies the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

(b) Use as 3 V output port

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

<2> Pull up externally the P72 pin (on-chip pull-up resistor cannot be used).

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

<4> Set the corresponding bit of the POM7 register to 1 to set the N-ch open drain output (V_{DD} withstand voltage) mode.

<5> Set the output mode by manipulating the PM7 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 LIN-UART or CAN controller.

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 4-8.

Remark The port pins mounted depend on the product. See **Table 4-3 Port Functions**.

Table 4-8. Settings of Port Mode Register and Output Latch When Using Alternate Function (1/5)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P00	INTP7	Input	1	×
	TI05	Input	1	×
	TO05	Output	0	0
P01	TI04	Input	1	×
	TO04	Output	0	0
P02	TI06	Input	1	×
	TO06	Output	0	0
P10	TI00	Input	1	×
	TO00	Output	0	0
	LTxD1	Output	0	1
	CTxD	Output	0	1
	$\overline{\text{SCK10}}$	Input	1	×
		Output	0	1
P11	TI02	Input	1	×
	SI10	Input	1	×
	CRxD	Input	1	×
	LRxD1	Input	1	×
	INTPLR1	Input	1	×
	TO02	Output	0	0
P12	INTP3	Input	1	×
	TI16	Input	1	×
	SO10	Output	0	1
	TO16	Output	0	0
P13	TI04	Input	1	×
	LTxD0	Output	0	1
	TO04	Output	0	0

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark	×:	don't care
	PM××:	Port mode register
	P××:	Port output latch

Table 4-8. Settings of Port Mode Register and Output Latch When Using Alternate Function (2/5)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P14	TI06	Input	1	×
	LRxD0	Input	1	×
	INTPLR0	Input	1	×
	TO16	Output	0	0
P15	TI10	Input	1	×
	SO00	Output	0	1
	TO10	Output	0	0
P16	TI12	Input	1	×
	SI00	Input	1	×
	TO12	Output	0	0
P17	TI14	Input	1	×
	$\overline{\text{SCK00}}$	Input	1	×
		Output	0	1
	TO14	Output	0	0
P30	INTP2	Input	1	×
	$\overline{\text{SSI00}}$	Input	1	×
	TI01	Input	1	×
	TO01	Output	0	0
P31	INTP2	Input	1	×
	TI11	Input	1	×
	STOPST	Output	0	0
	TO11	Output	0	0
P32	INTP4	Input	1	×
	TI13	Input	1	×
	TO13	Output	0	0
P40	TOOL0	I/O	×	×
	TI05	Input	1	×
	TO05	Output	0	0
P41	TOOL1	Output	0	1
	TI07	Input	1	×
	TO07	Output	0	0
P42	TxD2	Output	0	1
	SCL20	Output	0	1

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark ×: don’t care

PM××: Port mode register

P××: Port output latch

Table 4-8. Settings of Port Mode Register and Output Latch When Using Alternate Function (3/5)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P43	RxD2	Input	1	×
	SDA20	Input	1	×
	INTPR2	Input	1	×
	SDA20	Output	0	1
P44	TI07	Input	1	×
	TO07	Output	0	0
P45	TI10	Input	1	×
	TO10	Output	0	0
P46	TI12	Input	1	×
	TO12	Output	0	0
P47	INTP8	Input	1	×
P50	INTP3	Input	1	×
	TI20	Input	1	×
	TO20	Output	0	0
P51	TI21	Input	1	×
	TO21	Output	0	0
P52	TI22	Input	1	×
	STOPST	Output	0	0
	TO22	Output	0	0
P53	TI23	Input	1	×
	TO23	Output	0	0
P54	TI11	Input	1	×
	TO11	Output	0	0
P55	TI13	Input	1	×
	TO13	Output	0	0
P56	TI15	Input	1	×
	TO15	Output	0	0
P57	TI17	Input	1	×
	TO17	Output	0	0
P60	SCK00	Input	1	×
		Output	0	1
	SCL11	Output	0	1
P61	SI00	Input	1	×
	SDA11	I/O	0	1
P62	SO00	Output	0	1

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark	×:	don't care
	PM××:	Port mode register
	P××:	Port output latch

Table 4-8. Settings of Port Mode Register and Output Latch When Using Alternate Function (4/5)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P63	$\overline{\text{SSI00}}$	Input	1	×
P64	TI14	Input	1	×
	TO14	Output	0	0
P65	TI16	Input	1	×
	TO16	Output	0	0
P66	TI00	Input	1	×
	TO00	Output	0	0
P67	TI02	Input	1	×
	TO02	Output	0	0
P70	INTP5	Input	1	×
	KR0	Input	1	×
	TI15	Input	1	×
	TO15	Output	0	0
	LVIOUT	Output	0	0
P71	INTP6	Input	1	×
	KR1	Input	1	×
	TI17	Input	1	×
	TO17	Output	0	0
P72	KR2	Input	1	×
	CTxD	Output	0	1
	LTxD1	Output	0	1
P73	KR3	Input	1	×
	CRxD	Input	1	×
	LRxD1	Input	1	×
	INTPLR1	Input	1	×
P74	KR4	Input	1	×
	SO01	Output	0	1
P75	KR5	Input	1	×
	SI01	Input	1	×
P76	KR6	Input	1	×
	$\overline{\text{SCK01}}$	Input	1	×
		Output	0	1
P77	KR7	Input	1	×
	$\overline{\text{SSI01}}$	Input	1	×

Caution **The shaded pins are provided at two ports. Select either port by using the corresponding register.**

Remark ×: don't care

PM××: Port mode register

P××: Port output latch

Table 4-8. Settings of Port Mode Register and Output Latch When Using Alternate Function (5/5)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P80 to P87[Note]	ANI00 to ANI07[Note]	Input	1	×
P90 to P97[Note]	ANI08 to ANI15[Note]	Input	1	×
P100 to P107[Note]	ANI16 to ANI23[Note]	Input	1	×
P120	INTP0	Input	1	×
	EXLVI	Input	1	×
	TI11	Input	1	×
	TO11	Output	0	0
<R> P121	X1	–	1	×
<R> P122	X2	–	1	×
	EXCLK	Input	1	×
P124	EXCLKS	Input	1	×
P125	INTP1	Input	1	×
	ADTRG	Input	1	×
	TI03	Input	1	×
	TO03	Output	0	0
P126	TI01	Input	1	×
	TO01	Output	0	0
P127	TI03	Input	1	×
	TO03	Output	0	0
<R> P130	RESOUT	Output	×	×
P140	PCL	Output	0	0
P151	SO11	Output	0	1
P152	SI11	Input	1	×
P153	$\overline{\text{SCK11}}$	Input	1	×
		Output	0	1
P154	TI24	Input	1	×
	TO24	Output	0	0
P155	TI25	Input	1	×
	TO25	Output	0	0
P156	TI26	Input	1	×
	TO26	Output	0	0
P157	TI27	Input	1	×
	TO27	Output	0	0

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

Note The functions of the ANI00/P80 to ANI07/P87 and ANI08/P90 to ANI15/P97, and ANI16/P100 to ANI23/P107 pins can be selected by using the A/D port configuration register (ADPC), analog input channel specification register (ADS), and port mode registers 8 to 10 (PM8 to PM10).

ADPC Register	PM8 to PM10 Registers	ADS Register	ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97, ANI16/P100 to ANI23/P107 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark ×: don’t care
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 P10 is an output port, P11 to P17 are input ports (all pin statuses are high level), and the port latch value of port 1 is 00H, if the output of output port P10 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 1 is FFH.

Explanation: The targets of writing to and reading from the Pn register of a port whose PMn_m 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/Hx3.

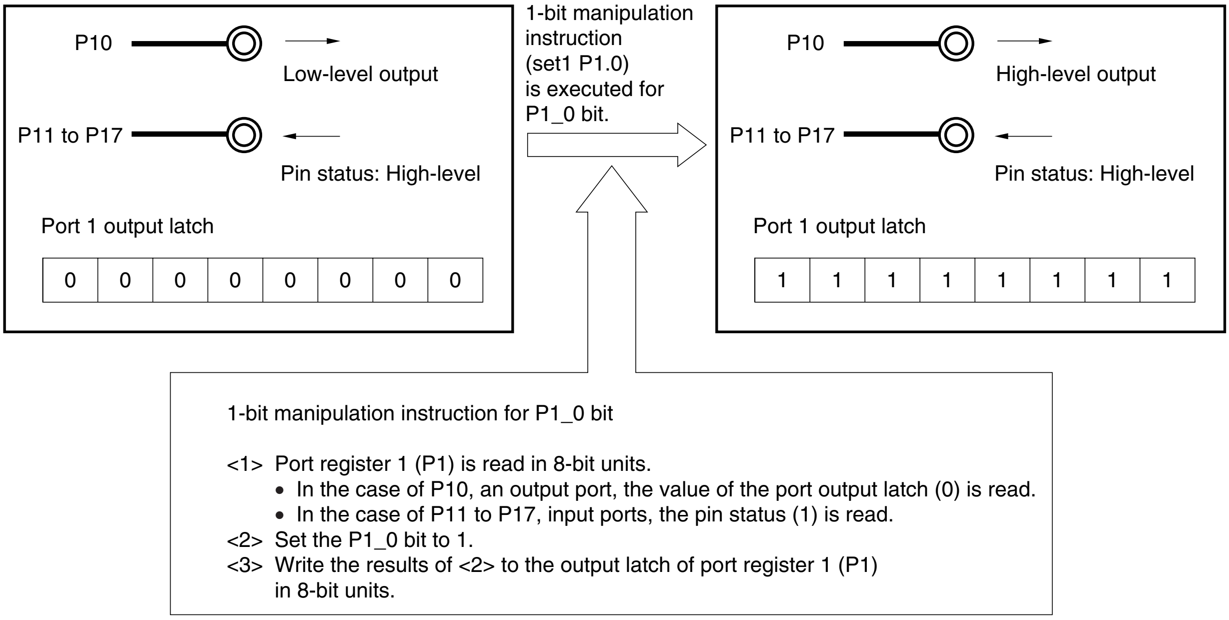
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P17, which are input ports, are read. If the pin statuses of P11 to P17 are high level at this time, the read value is FEH.

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

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

Figure 4-93. Bit Manipulation Instruction (P10)



1-bit manipulation instruction for P1_0 bit

<1> Port register 1 (P1) is read in 8-bit units.
- In the case of P10, an output port, the value of the port output latch (0) is read.
- In the case of P11 to P17, input ports, the pin status (1) is read.

<2> Set the P1_0 bit to 1.
<3> Write the results of <2> to the output latch of port register 1 (P1) in 8-bit units.

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

This circuit oscillates a clock of f_{IH} = 4 or 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 of HIOSTOP (bit 0 of CSC).

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)).

(2) PLL clock

A clock that is the main system clock multiplied by 1, 6, or 8 can be oscillated. Oscillation can be stopped by executing a STOP instruction or by setting PLLON (bit 0 of PLLCTL).

(3) Subclock

An external subclock (f_{EXS}) can be supplied as the subclock (f_{SUB}) from the EXCLKS/P124 pin. The subclock cannot be used as the CPU clock. The wakeup timer is the only hardware operating on the subclock. External subclock input can be disabled by setting XTSTOP.

Remarks 1. The oscillation frequency of the internal high-speed oscillation clock can be selected as 4 MHz or 8 MHz by setting bit 1 (SEL4M) of the option byte (000C1H). See **CHAPTER 23 OPTION BYTE** for details of setting the option byte.

2. The multiplication number of the PLL clock is set by using bit 2 (OPTPLL) of the option byte (000C1H). See **CHAPTER 23 OPTION BYTE** for details of setting the option byte.

3. f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency
f_{EXS}: External subclock frequency
f_{SUB}: Subclock frequency

(4) Internal low-speed oscillation clock

- **Internal low-speed oscillator**

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

  Oscillation operation of the internal low-speed oscillator after reset release can be controlled by setting bit 7 (LIOUSE) of the option byte (000C1H).

  If operation of the internal low-speed oscillator has been enabled by using bit 7 (LIOUSE) of the option byte (000C1H) after reset release, the internal low-speed oscillator starts to oscillate automatically.

  When bit 7 (LIOUSE) and bit 5 (LIOSYSB) of the option byte (000C1H) are “1” and “0”, respectively, The internal low-speed oscillation clock can be used as the CPU or peripheral hardware clock by setting CSS (bit 6 of the system clock control register (CKC)) .

  Furthermore, when bit 6 (LIOSTOPB) of the option byte (000C1H) is “0”, respectively, oscillation can be stopped by executing a STOP instruction.

Caution Executing a STOP instruction is prohibited when the internal low-speed oscillation clock is selected as the CPU or peripheral hardware clock (CSS = 1).

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

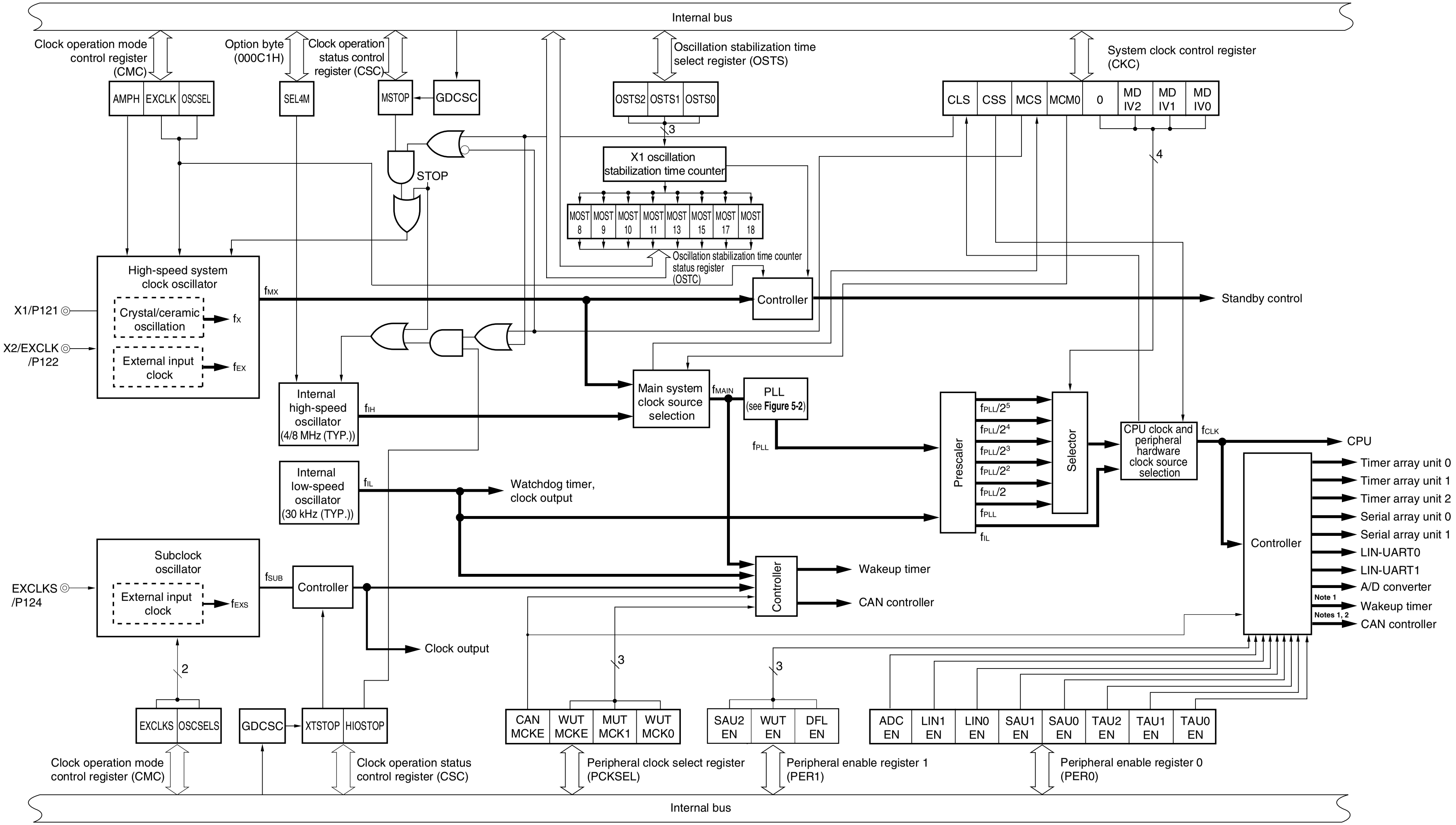
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) System clock control register (CKC) Peripheral enable registers 0, 1 (PER0, PER1) Peripheral clock select register (PCKSEL) Operation speed mode control register (OSMC) PLL control register (PLLCTL) PLL status register (PLLSTS)
Oscillators	X1 oscillator Subclock input oscillator Internal high-speed oscillator Internal low-speed oscillator

Figure 5-1. Block Diagram of Clock Generator



(**Notes 1, 2** and **Remark** are given on the next page.)

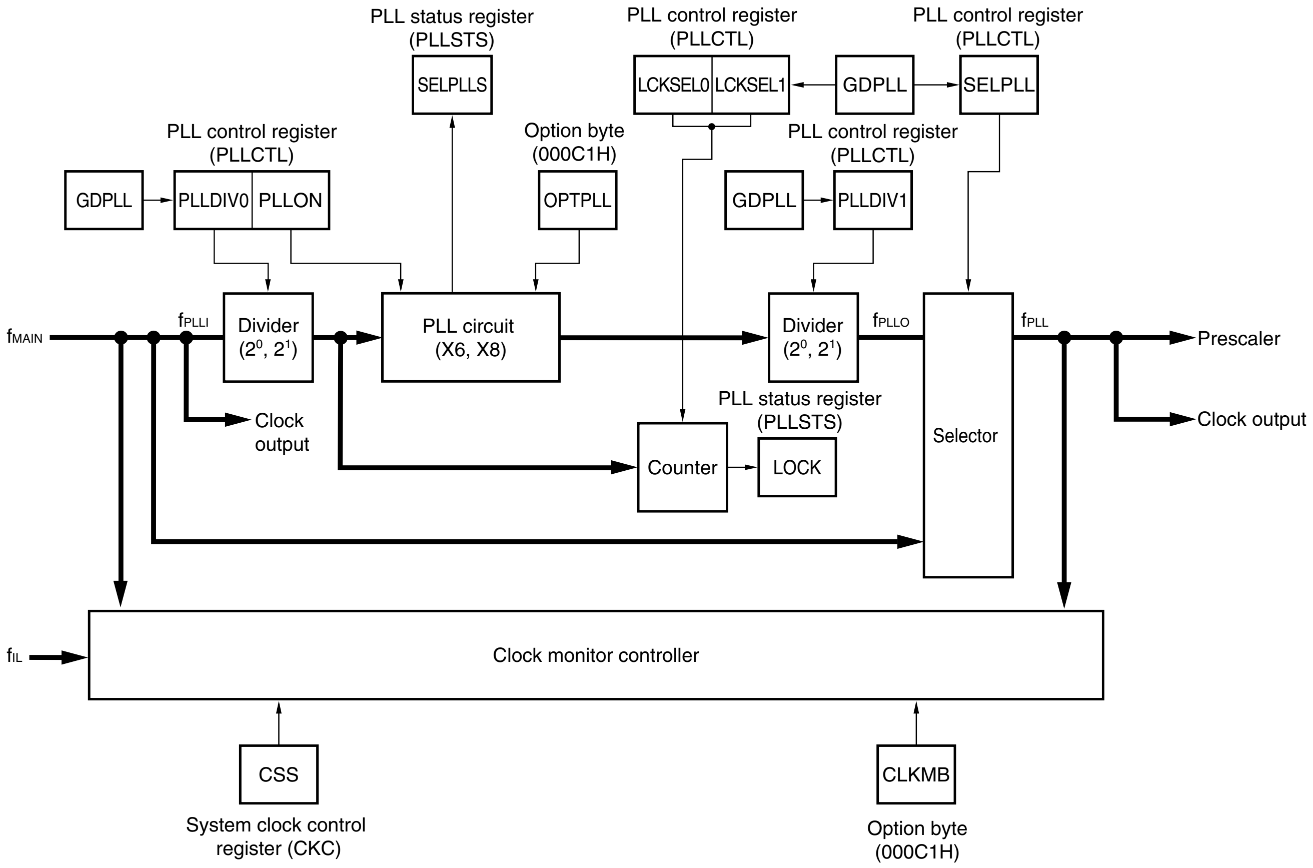
Notes 1. Can be used only for SFR access.

2. Only the main system clock (f_{MAIN}) can be selected as the CAN controller operating clock.

Remark

f_X:	X1 clock
f_{IH}:	Internal high-speed oscillation clock
f_{EX}:	External main system clock
f_{MX}:	High-speed system clock
f_{MAIN}:	Main system clock
f_{EXS}:	External subclock
f_{SUB}:	Subclock
f_{CLK}:	CPU/peripheral hardware clock
f_{IL}:	Internal low-speed oscillation clock

Figure 5-2. Block Diagram of PLL Circuit



Remark

f_{MAIN}:	Main system clock
f_{IL}:	Internal low-speed oscillation clock
f_{PLLI}:	PLL can be input clock
f_{PLLO}:	PLL output clock
f_{PLL}:	PLL clock

5.3 Registers Controlling Clock Generator

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

- Clock operation mode control register (CMC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- System clock control register (CKC)
- Peripheral enable registers 0, 1 (PER0, PER1)
- Peripheral clock select register (PCKSEL)
- Operation speed mode control register (OSMC)
- PLL control register (PLLCTL)
- PLL status register (PLLSTS)

(1) Clock operation mode control register (CMC)

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

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 5-3. Format of Clock Operation Mode Control Register (CMC)

Address: FFFA0H After reset: 00H R/W

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

EXCLKS	OSCSELS	Subclock pin operation mode	EXCLKS/P124 pin
0	0	Input port mode	Input port
0	1	Setting prohibited	
1	0	Input port mode	Input port
1	1	External clock input mode	External clock input

AMPH	Control of X1 clock oscillation frequency
0	$2\ \text{MHz} \le f_X \le 10\ \text{MHz}$
1	$2\ \text{MHz} < f_X \le 20\ \text{MHz}$

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).

3. Be sure to set AMPH to 1 if the X1 clock oscillation frequency exceeds 10 MHz. Note that this setting is unnecessary when using the external clock input mode.

4. It is recommended to set the default value (00H) to CMC after reset release, even when the register is used at the default value, in order to prevent malfunctioning during a program loop.

Remark f_{SUB}: Subclock frequency

f_{EXS}: External subclock frequency

f_X: X1 clock frequency

(2) Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock and internal high-speed oscillation clock (except the 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.

Caution When writing to the CSC register, set the GDCSC bit of the GUARD register to “1”.

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	Subclock operation control	
	External clock input mode	Input port mode
0	External clock from EXCLKS pin is valid	–
1	External clock from EXCLKS 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.

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

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

4. The oscillation frequency of the internal high-speed oscillation clock (f_{IH}) can be selected by using bit 1 (SEL4M) of the option byte (000C1H).

5. 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	• CLS = 0 and MCS = 0 • CLS = 1 (CPU and peripheral hardware clocks operate with a clock other than the high-speed system clock.)	MSTOP = 1
External main system clock		
Internal high-speed oscillation clock	• CLS = 0 and MCS = 1 • CLS = 1 (CPU and peripheral hardware clocks operate with a clock other than the internal high-speed oscillation clock.)	HIOSTOP = 1

(3) 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 internal low-speed oscillation 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

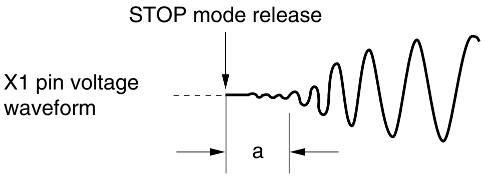
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 8 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	32 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	32 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	64 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	128 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	256 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	1.02 ms min.	409.6 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	4.10 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	16.38 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	32.77 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 or equal to the count value which is to be checked by the OSTC register.

- **If the X1 clock starts oscillation while the internal high-speed oscillation clock or internal low-speed oscillation 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

(4) 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		
				fx = 8 MHz	fx = 20 MHz
0	0	0	2^{8}/fx	32 μs	Setting prohibited
0	0	1	2^{9}/fx	64 μs	25.6 μs
0	1	0	2^{10}/fx	128 μs	51.2 μs
0	1	1	2^{11}/fx	256 μs	102.4 μs
1	0	0	2^{13}/fx	1.02 ms	409.6 μs
1	0	1	2^{15}/fx	4.10 ms	1.64 ms
1	1	0	2^{17}/fx	16.38 ms	6.55 ms
1	1	1	2^{18}/fx	32.77 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. Use the OSTS register to set an appropriate oscillation stabilization time, before using the MSTOP bit to oscillate the X1 clock.

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

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

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

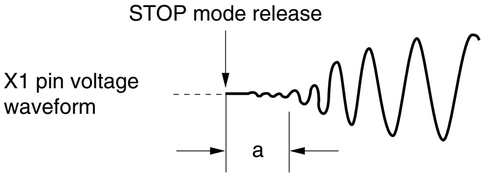
6. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS.

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

- **If the X1 clock starts oscillation while the internal high-speed oscillation clock or internal low-speed oscillation 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.)**

7. If the STOP mode is released when the PLL clock is selected as the CPU clock, the time of the oscillation stabilization time set using OSTS plus the PLL lockup wait time will be required.

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



Remark fx: X1 clock oscillation frequency

(5) System clock control register (CKC)

This register is used to select the main system clock and CPU/peripheral hardware clock and set a division ratio.
CKC can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 01H.

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

Address: FFFA4H After reset: 01H R/W[Note 1]

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
CKC	CLS	CSS[Note 2]	MCS	MCM0[Notes 2, 3]	0	MDIV2	MDIV1	MDIV0

CLS	Status flag of CPU/peripheral hardware clock (f_{CLK})
0	PLL clock (f_{PLL}) being selected
1	Internal low-speed oscillation clock (f_{IL}) being selected

CSS[Note 2]	Bit to control selection of CPU/peripheral hardware clock (f_{CLK})
0	PLL clock/Clock through (f_{PLL})
1	Internal low-speed oscillation clock (f_{IL})

MCS	Status flag of main system clock (f_{MAIN})
0	Internal high-speed oscillation clock (f_{IH}) being selected
1	High-speed system clock (f_{MX}) being selected

MCM0[Notes 2, 3]	Bit to control selection of main system clock (f_{MAIN})
0	Internal high-speed oscillation clock (f_{IH})
1	High-speed system clock (f_{MX})

<R>

CLS	MDIV2	MDIV1	MDIV0	Division of PLL clock (f_{PLL})
0	0	0	0	f_{PLL}
	0	0	1	$f_{PLL}/2$
	0	1	0	$f_{PLL}/2^2$
	0	1	1	$f_{PLL}/2^3$
	1	0	0	$f_{PLL}/2^4$
	1	0	1	$f_{PLL}/2^5$ [Note 4]
	Other than above			Setting prohibited
1	X	X	X	f_{IL}

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

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

3. Changing the value of the MCM0 bit is prohibited while PLLON is set to 1.

4. Setting prohibited if $f_{PLL} < 4$ MHz

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

(Cautions 2, 3, and remark are given on the next page.)

Cautions 2. If the internal low-speed oscillation clock is used as the peripheral hardware clock, the operation of the A/D converter is not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS.

3. The clock set by using CSS, MCM0, 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 watchdog timer, clock output, 16-bit wakeup timer, and CAN controller) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral hardware clock.

Remark For setting the PLL clock, see **(11) PLL control register (PLLCTL)** and **5.7.3 (1) Example of setting procedure when oscillating PLL clock.**

(6) Peripheral enable registers 0, 1 (PER0, PER1)

These registers are 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 and PER1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears theses registers to 00H.

Caution Whether to enable or disable SFR writing only is selected for the 16-bit wakeup timer.
Whether to enable or disable supplying the operating clock is selected using the PCKSEL register.

Figure 5-8. Format of Peripheral Enable Registers 0, 1 (PER0, PER1) (1/3)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	ADCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN

Address: F00F1H After reset: 00H R/W

Symbol	7	6	5	4	<3>	2	<1>	<0>
PER1	0	0	0	0	SAU2EN	0	WUTEN	DFLEN

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

LIN1EN	Control of asynchronous serial interface LIN-UART1 input clock
0	Stops input clock supply. • SFR used by LIN-UART1 cannot be written. • LIN-UART1 is in the reset status.
1	Supplies input clock. • SFR used by LIN-UART1 can be read and written.

LIN0EN	Control of asynchronous serial interface LIN-UART0 input clock
0	Stops input clock supply. • SFR used by LIN-UART0 cannot be written. • LIN-UART0 is in the reset status.
1	Supplies input clock. • SFR used by LIN-UART0 can be read and written.

Caution Be sure to clear the following bits to 0.
78K0R/HC3: Bit 2 of the PER0 register, bits 2 to 7 of the PER1 register
78K0R/HE3, 78K0R/HF3, 78K0R/HG3: Bits 2, 4 to 7 of the PER1 register

Figure 5-8. Format of Peripheral Enable Registers 0, 1 (PER0, PER1) (2/3)

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

SAU0EN	Control of serial array unit 0 input clock
0	Stops input clock supply. • 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 and written.

TAU2EN	Control of timer array unit 2 input clock
0	Stops input clock supply. • SFR used by timer array unit 2 cannot be written. • Timer array unit 2 is in the reset status.
1	Supplies input clock. • SFR used by timer array unit 2 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.

SAU2EN	Control of serial array unit 2 input clock
0	Stops input clock supply. • SFR used by serial array unit 2 cannot be written. • Serial array unit 2 is in the reset status.
1	Supplies input clock. • SFR used by serial array unit 2 can be read and written.

Caution Be sure to clear the following bits to 0.
78K0R/HC3: Bit 2 of the PER0 register, bits 2 to 7 of the PER1 register
78K0R/HE3, 78K0R/HF3, 78K0R/HG3: Bits 2, 4 to 7 of the PER1 register

Figure 5-8. Format of Peripheral Enable Registers 0, 1 (PER0, PER1) (3/3)

<R>

WUTEN	Control of 16-bit wakeup timer input clock
0	Stops input clock supply for SFR writing[Note]. • SFR used by the 16-bit wakeup timer can be read and written.
1	Supplies input clock for SFR writing. • SFR used by the 16-bit wakeup timer can be written.

<R> **Note** Even if it stops this clock supply, clock supply of operation does not stop.
Please stop clock supply of operation by a PCKSEL register to decrease power consumption.

DFLEN	Control of data flash input clock
0	Stops input clock supply. • Cannot be read and written for data flash area.
1	Supplies input clock. • Can be read and written for data flash area.

Caution Be sure to clear the following bits to 0.
78K0R/HC3: Bit 2 of the PER0 register, bits 2 to 7 of the PER1 register
78K0R/HE3, 78K0R/HF3, 78K0R/HG3: Bits 2, 4 to 7 of the PER1 register

(7) Peripheral clock select register (PCKSEL)

This register is used to select for and supply to each peripheral hardware device the operating clock.

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

Reset signal generation clears this register to 00H.

Caution Set the PCKSEL register before starting to operate each peripheral hardware device.

Figure 5-9. Format of Peripheral Clock Select Register (PCKSEL)

Address: F00F2H After reset: 00H R/W

Symbol	7	6	5	<4>	3	<2>	1	0
PCKSEL	0	0	0	CANMCKE	0	WUTMCKE	WUTMCK1	WUTMCK0

CANMCKE	Control of CAN controller operating clock[Note]
0	Stops supplying operating clock.
1	Supplies operating clock.

WUTMCKE	Control of 16-bit wakeup timer operating clock
0	Stops supplying operating clock.
1	Supplies operating clock.

WUTMCK1	WUTMCK0	16-bit wakeup timer operating clock selection
0	0	f_{IL}
0	1	f_{SUB}
1	0	$f_{MAIN}/2^8$
1	1	$f_{MAIN}/2^{12}$

Note This bit controls the main system clock (f_{MAIN}) that is supplied to the CAN controller operating clock. As the main system clock, the high-speed system clock (f_{MX}) or internal high-speed oscillation clock (f_{IH}) can be selected. Do not set CANMCKE to 1 with both the high-speed system clock (f_{MX}) and internal high-speed oscillation clock (f_{IH}) stopped.

Cautions 1. Be sure to clear bits 3, and 5 to 7 of the PCKSEL register to 0.

<R> **2. Please change WUTMCK1 and WUTMCK0 after stopping a 16-bit wake up timer (WUTMCKE = 0).**

(8) Operation speed mode control register (OSMC)

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

If the microcontroller operates at a low speed with a system clock of no more than 20 MHz or 10 MHz, the power consumption can be lowered by setting this register to the default value, 00H.

OSMC can be set by a 1-bit/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	0	0	0	0	0	DMSTP	0	FSEL

DMSTP	Selection the clock frequency (f_{CLK}) of data flash
0	Operates at a frequency of 20 MHz or less (default).
1	Operates at a frequency higher than 20 MHz.

FSEL	Selection the clock frequency (f_{CLK}) of code flash
0	Operates at a frequency of 10 MHz or less (default).
1	Operates at a frequency higher than 10 MHz.

Cautions

1. **Stop the DMA controller before writing “1” to FSEL.**
2. **The CPU waits when “1” is written to the FSEL flag.**
   The wait time is 20 μs ±3 clocks (Max.).
   However, counting the oscillation stabilization time of f_X can continue even while the CPU is waiting.
3. **When switching f_{CLK} to at least 10 MHz, do so after setting FSEL to 1 and waiting for at least 3 clocks to elapse.**
4. **Operation can be performed at a frequency of no more than 20 MHz or 10 MHz, even if DMSTP = 1 or FSEL = 1.**
5. **Disable interrupt when setting FSEL to “1”. Enable interrupt after at least two clocks have elapsed since FSEL was set to “1”.**
6. **After setting FSEL to 1, wait for at least 3 clock cycles before enabling DMA.**
7. **Rewriting DMSTP is prohibited when DFLEN = 1.**
8. **After setting FSEL to 1, wait for at least 3 clock cycles before accessing the CAN registers or data flash memory.**

(9) PLL control register (PLLCTL)

This 8-bit register is used to control the PLL function.

A clock that is the main system clock multiplied by 1, 6, or 8 can be selected as the CPU clock or peripheral hardware clock.

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

Reset signal generation clears this register to 00H.

Caution Set the GDPLL bit of the GUARD register to 1 when writing to the PLLCTL register.

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

Address: F007H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PLLCTL	LCKSEL1	LCKSEL0	PLLDIV1	PLLDIV0	0	SELPLL	0	PLLON

<R>

LCKSEL1	LCKSEL0	Lockup wait counter setting value
0	0	$2^8/f_{MAIN}/2^{PLLDIV0}$
0	1	$2^9/f_{MAIN}/2^{PLLDIV0}$
1	0	$2^{10}/f_{MAIN}/2^{PLLDIV0}$
1	1	Setting prohibited

<R>

PLLDIV1	PLL output clock (f_{PLLO}) selection
0	If $f_{MAIN}/2^{PLLDIV0}$ × PLL multiplication numbers ≤ 24 MHz
1	If $f_{MAIN}/2^{PLLDIV0}$ × PLL multiplication numbers > 24 MHz

PLLDIV0	PLL input clock (f_{PLLI}) selection
0	When f_{MAIN} = 4 MHz
1	When f_{MAIN} = 8 MHz

SELPLL	Clock mode selection
0	Clock through mode (f_{MAIN})
1	PLL clock select mode (f_{PLLO})

PLLON	PLL operation control
0	Stops PLL
1	Operates PLL (A lockup wait time is required after the PLL starts operating, so that the frequency stabilizes.)

(**Cautions 1 to 9**, and **Remarks 1 to 3** are given on the next page.)

Cautions 1. When the PLL output is not stable (LOCK = 0), writing to the SELPLL bit is prohibited.

2. When the clock monitor function detects that the PLL clock is stopped, the SELPLL bit is not automatically cleared.
When the clock monitor function detects that the PLL clock is stopped, the PLLSTS.SELPLLS is automatically cleared.

3. Select a time of at least 100 μs for the lockup wait time counter.

4. When the PLL starts operating, a wait time until the PLL is locked is required.

5. Only the combinations of PLL input clocks and multiplication numbers shown in the following table are available when using a PLL. Any input clock of a frequency of 2 to 20 MHz can be selected when not using a PLL (PLLON = 0 or SELPLL = 0).

Option Byte (000C1H)	PLLCTL Register		Frequency That Can Be Input (f_{MAIN})	Output Frequency (f_{PLL})
OPTPLL	PLLDIV1	PLLDIV0		
0	1	0	4 MHz ±2%	16 MHz ±2%
0	1	1	8 MHz ±2%	16 MHz ±2%
1	0	0	4 MHz ±2%	24 MHz ±2%
1	0	1	8 MHz ±2%	24 MHz ±2%
Other than above			Setting prohibited	

6. When PLLON = 0, simultaneous rewriting by accessing the PLLON and SELPLL bits in 8-bit units is prohibited.

7. When the PLLON bit is cleared (0), the SELPLL bit is also automatically cleared (clock through mode).

<R> **8. When PLLON = 1, rewriting of PLLDIV1-PLLDIV0 is prohibition.**

<R> **9. When PLLON = 1, changing of f_{MAIN} is prohibition.**

Remarks 1. If the PLLON and SELPLL bits are set, the clock selected for fPLL will be determined according to the states of LOCK and SELPLLS. The clocks selected for fPLL when the PLLON, SELPLL, LOCK, and SELPLLS bits are set are shown below.

PLLON	SELPLL	LOCK	SELPLLS	Selection clock (f_{PLL})
0	0	0	0	Main system clock (f_{MAIN})
1	0	0	0	Main system clock (f_{MAIN})
1	0	1	0	Main system clock (f_{MAIN})
1	1	1	0	Main system clock (f_{MAIN}) Not switched to multiplication clock after setting SELPLL = 1
1	1	1	1	Clock that is a multiple of the main system clock (f_{MAIN})
Other than above				Setting prohibited

2. The PLL multiplication number is set by using bit 2 (OPTPLL) of the option byte (000C1H). See **CHAPTER 23 OPTION BYTE** for details.

3. f_{PLLI} : PLL input clock frequency
f_{PLL} : PLL clock frequency
f_{MAIN} : Main system clock frequency

(10) PLL status register (PLLSTS)

This register indicates the operating state of the PLL clock.

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

Reset signal generation clears this register to 00H.

Figure 5-12. Format of PLL Status Register (PLLSTS)

Address: F00F6H After reset: 00H R

Symbol	<7>	6	5	4	<3>	2	1	0
PLLSTS	LOCK	0	0	0	SELPLLS	0	0	0

LOCK	PLL lock state
0	Unlocked state
1[Note]	Locked state

SELPLLS	State of the clock mode
0	Clock through mode (f_{MAIN})
1	PLL clock select mode (f_{PLL})

Note This is set (1) when the lockup wait counter overflows.

Caution When the PLL starts operating, a wait time until the PLL is locked (LOCK = 1) is required.

Remark The PLL multiplication number is set by using bit 2 (OPTPLL) of the option byte (000C1H). See **CHAPTER 23 OPTION BYTE** for details.

5.4 Clock Monitor

The clock monitor uses the internal low-speed oscillator to sample the main system clock (f_{MAIN}) and PLL clock (f_{PLL}). If oscillation of the main system clock stops, a reset request signal (RESCLM) is generated. If the PLL clock is stopped, clock through mode is selected by default and SELPLLS (but not SELPLL) is cleared. An interrupt request signal (INTCLM) is also generated at this time.

(1) Configuration

A block diagram of the clock monitor is shown below.

Figure 5-13. Clock Monitor Block Diagram

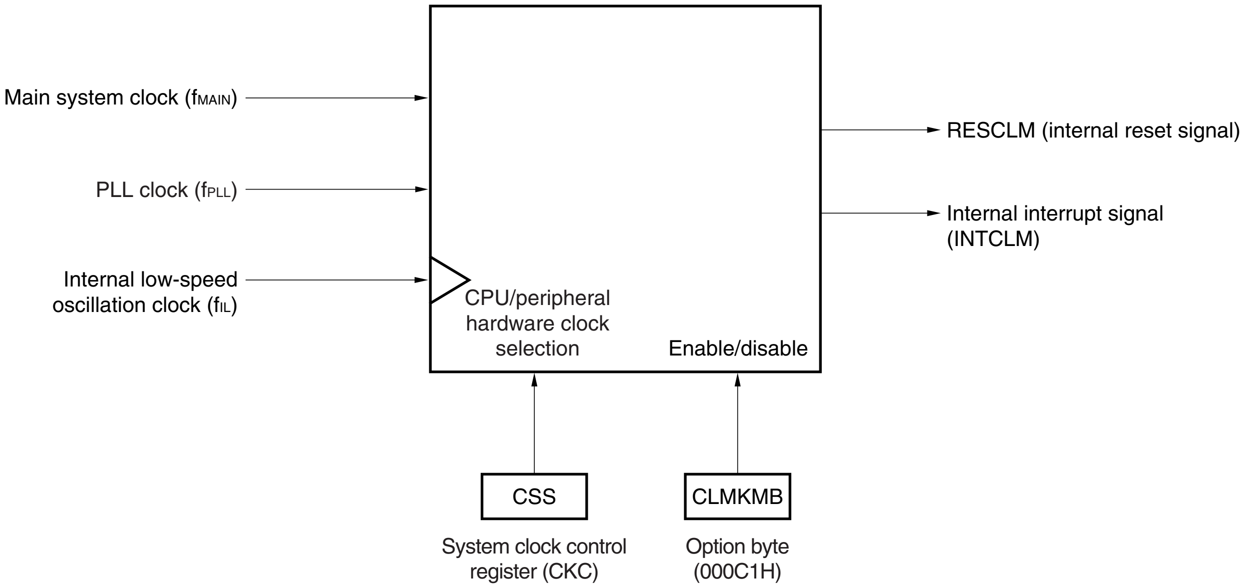

Main system clock (fMAIN)
PLL clock (fPLL)
Internal low-speed oscillation clock (fIL)
CPU/peripheral hardware clock selection
Enable/disable
RESCLM (internal reset signal)
Internal interrupt signal (INTCLM)
CSS
System clock control register (CKC)
CLMKMB
Option byte (000C1H)

Table 5-3. Clock Monitor Operation States (CLMKMB Bit = 0)

CPU/Peripheral Hardware Clock (f_{CLK})	Operation Mode	Main Clock Oscillator State	Internal Low-Speed Oscillator State	Clock Monitor State
PLL clock (f_{PLL})	HALT mode	Oscillates	Oscillates	Operates Note 1
	STOP mode	Stopped	Oscillates	Stopped Note 2
Internal low-speed oscillation clock (f_{IL})	–	Stopped	Oscillates	Stopped
During reset	–	Stopped	Stopped	Stopped

Notes 1. The clock monitor also stops if the internal low-speed oscillator stops.

2. The clock monitor also stops when the oscillation stabilization time is counted after STOP mode is released.

(2) Operation starting and stopping

To enable the operation of the clock monitor, set bit 4 (CLKMB) of the option byte (000C1H) to 0.
Clock monitoring is automatically started after a reset is released.
The clock monitor stops automatically under the following conditions.

- When in STOP mode
- When counting the oscillation stabilization time after STOP mode is released
- When the CPU/peripheral hardware clock frequency (f_{CLK}) equals the internal low-speed oscillation clock (f_{IL})
- When the sampling clock is stopped (stop of internal low-speed oscillator)
- When bit 4 (CLKMB) of the option byte (000C1H) is 1

5.5 System Clock Oscillator

5.5.1 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.

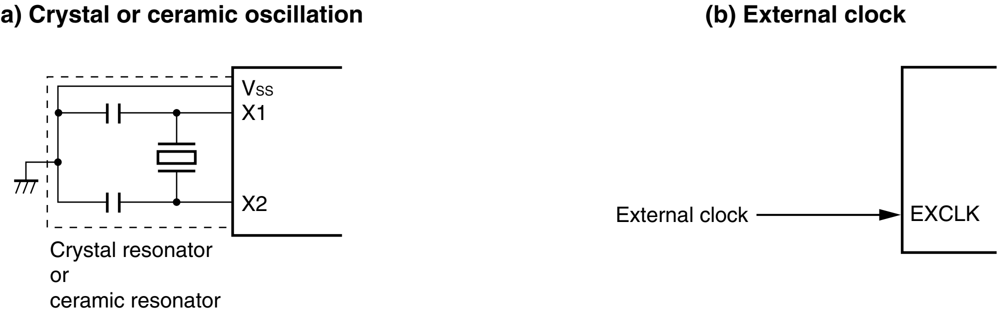
• Crystal or ceramic oscillation:	EXCLK, OSCSEL = 0, 1
• External clock input:	EXCLK, OSCSEL = 1, 1

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

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

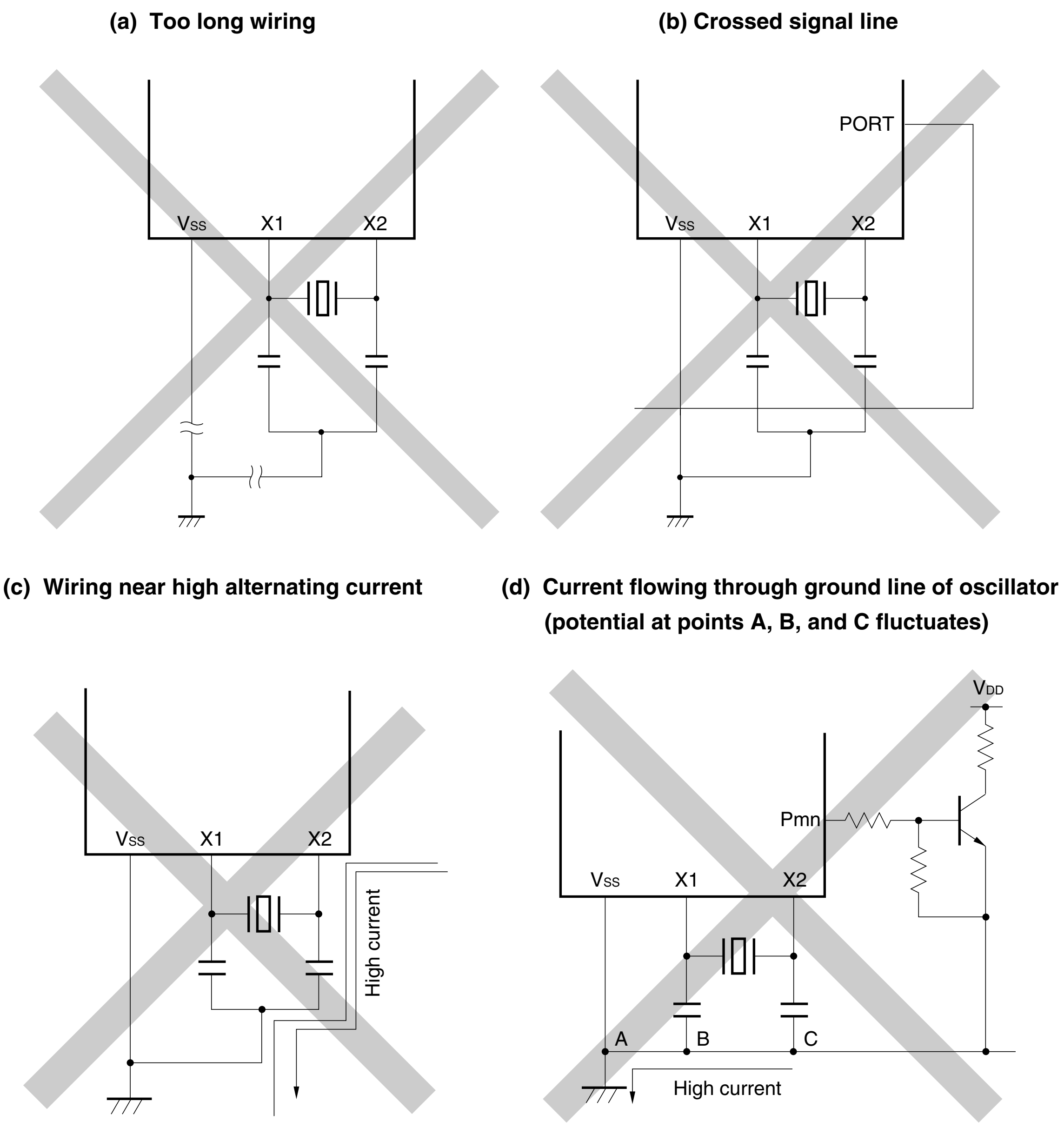


Caution When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in Figure 5-14 to avoid an adverse effect from wiring capacitance.

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

Figure 5-15 shows examples of incorrect resonator connection.

Figure 5-15. Examples of Incorrect Resonator Connection (1/2)



(a) Too long wiring

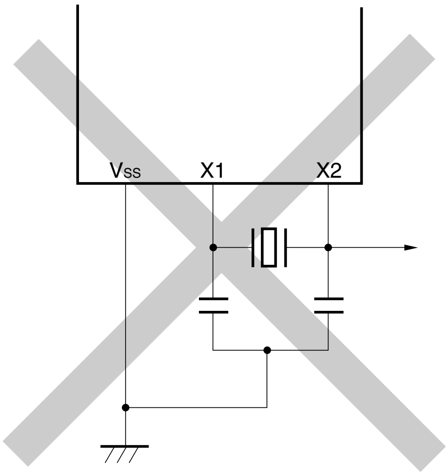
(b) Crossed signal line

(c) Wiring near high alternating current

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

Figure 5-15. Examples of Incorrect Resonator Connection (2/2)

(e) Signals are fetched

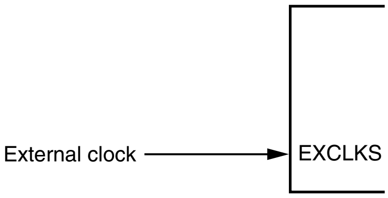


5.5.2 Subclock input oscillator

Subclock can be input to the subclock input oscillator. Input a clock signal to the EXCLKS pin when doing so.

Figure 5-16 shows an example of an external circuit of the subclock input oscillator.

Figure 5-16. Example of External Circuit of Subclock Input Oscillator



5.5.3 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0R/Hx3 (4, 8 MHz (TYP.)). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC).

The oscillation frequency of the internal high-speed oscillation clock can be selected as 4 MHz or 8 MHz by setting bit 1 (SEL4M) of the option byte (000C1H). See **CHAPTER 23 OPTION BYTE** for details of setting the option byte.

After a reset release, the internal high-speed oscillator automatically starts oscillation.

5.5.4 PLL circuit

The PLL circuit is incorporated in the 78K0R/Hx3. Operation of the PLL circuit can be controlled by using bit 0 (PLLON) of the PLL control register (PLLCTL). Set it to a multiple of the PLL clock (×6 and ×8 are set by using bit 2 (OPTPLL) of the option byte (000C1H). See **CHAPTER 23 OPTION BYTE** for details of setting the option byte.

5.5.5 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0R/Hx3 (30 kHz (TYP.)).

Oscillation operation of the internal low-speed oscillator after a reset release can be controlled by using bit 7 (LIOUSE) of the option byte (000C1H).

The internal low-speed oscillator starts to oscillate automatically, when its operation has been enabled by using bit 7 (LIOUSE) of the option byte (000C1H) after a reset release.

5.5.6 Prescaler

The prescaler generates the CPU/peripheral hardware clock by dividing the main system clock and internal low-speed oscillation clock.

5.6 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}
- Subclock f_{SUB}
  - External subclock f_{EXS}
- PLL clock f_{PLL}
- 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/Hx3, thus enabling the following.

(1) Enhancement of security function

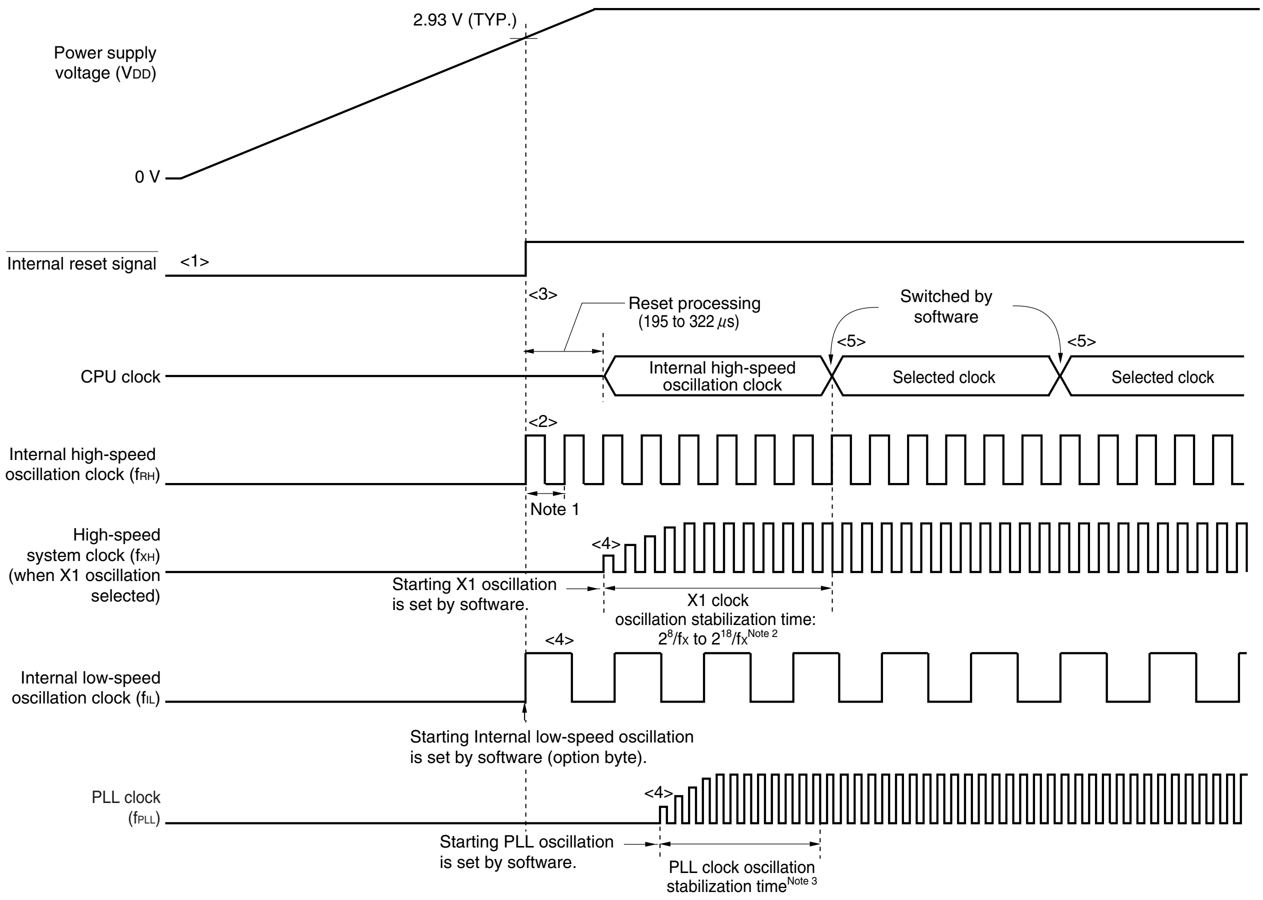
When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the internal high-speed oscillation clock, so the device can be started 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 5-17.

Figure 5-17. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Operation Enabled and Internal Low-Speed Oscillator Enable usage Are Set (Option Byte: LVIOFF = 0, LIOUSE = 1))


2.93 V (TYP.)
Power supply voltage (VDD)
0 V
Internal reset signal
<1>
<3>
Reset processing (195 to 322 μs)
Switched by software
<5>
<5>
CPU clock
Internal high-speed oscillation clock
Selected clock
Selected clock
<2>
Internal high-speed oscillation clock (fRH)
Note 1
High-speed system clock (fXH) (when X1 oscillation selected)
<4>
Starting X1 oscillation is set by software.
X1 clock oscillation stabilization time: 2^8/fx to 2^18/fx Note 2
<4>
Internal low-speed oscillation clock (fIL)
Starting Internal low-speed oscillation is set by software (option byte).
PLL clock (fPLL)
<4>
Starting PLL oscillation is set by software.
PLL clock oscillation stabilization time Note 3

<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 2.93 V (TYP.), the reset is released and the internal high-speed oscillator 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.

<4> Set the start of oscillation of the X1 or internal low-speed oscillation clock, or PLL clock via software (see **(1)** in **5.7.1 Example of controlling high-speed system clock**, **(1)** in **5.7.4 Example of controlling PLL clock**, and **5.7.5 Example of controlling internal low-speed clock)**.

<5> When switching the CPU clock to the X1 or internal low-speed oscillation clock, wait for the clock oscillation to stabilize, and then set switching via software (see **(3)** in **5.7.1 Example of controlling high-speed system clock**, and **5.7.3 Example of controlling internal low-speed clock**).

Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock and internal low-speed oscillation clock.

2. When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed system clock (X1 oscillation), set the oscillation stabilization time when releasing STOP mode using the oscillation stabilization time select register (OSTS).

3. A time until the PLL is locked (LOCK = 1) is required when starting to operate the PLL.

Cautions 1. A voltage oscillation stabilization time (about 2.1 to 5.8 ms) is required after the supply voltage reaches 1.61 V (TYP.). If the supply voltage rises from 1.61 V (TYP.) to 2.93 V (TYP.) within the power supply oscillation stabilization time, the power supply oscillation stabilization time is automatically generated before reset processing.

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

Remark While the microcontroller is operating, a clock that is not used as the CPU clock can be stopped via software settings. The internal high-speed oscillation clock and high-speed system clock can be stopped by executing the STOP instruction (see **(4)** in **5.7.1 Example of controlling high-speed system clock**, **(3)** in **5.7.2 Example of controlling internal high-speed oscillation clock**, **(2)** in **5.7.4 Example of controlling PLL clock**, and **5.7.5 Example of controlling internal Low-speed oscillation clock**).

5.7 Controlling Clock

5.7.1 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	0/1	0/1	0	0	0	0

- 10 MHz < fx ≤ 20 MHz

EXCLK	OSCSEL	EXCLKS	OSCSELS	0	0	0	AMPH
0	1	0/1	0/1	0	0	0	1

Remark For setting of the P124 pins, see **(1)** in **5.7.3 Example of controlling subclock**.

<2> Enabling CSC manipulation
Manipulating CSC is enabled by setting GDCSC to 1.

<3> Controlling oscillation of X1 clock (CSC register)
If MSTOP is cleared to 0, the X1 oscillator starts oscillating.

<4> 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 5.7.3 Example of controlling subclock.

2. Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 29 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	0/1	0/1	0	0	0	×

Remarks 1. ×: don't care

2. For setting of the P124 pins, see **5.7.3 (1) Example of setting procedure when using an external sub-clock**.

<2> Enabling CSC manipulation
Manipulating CSC is enabled by setting GDCSC to 1.

<3> 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 bit at the same time. For OSCSELS bit, see 5.7.3 Example of controlling subclock.

2. Set the external main system clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 29 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 and selecting the high-speed system clock as the main system clock (MCM0 = 1)[Note]
(See **5.7.1 (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 PLL clock oscillation[Note]
(See **5.7.4 (1) Example of setting procedure when oscillating PLL clock** or **(2) Example of setting procedure when stopping PLL clock**.)

Note The setting of <2> is not necessary when the PLL is not used.

<3> Setting the PLL clock as 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_{PLL}
	0	0	1	$f_{PLL}/2$
	0	1	0	$f_{PLL}/2^2$
	0	1	1	$f_{PLL}/2^3$
	1	0	0	$f_{PLL}/2^4$
	1	0	1	$f_{PLL}/2^5$ [Note]

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

<4> If some peripheral hardware macros are not used, supply of the input clock to each hardware macro can be stopped.

(PER0 register)

ADCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN

(PER1 register)

0	0	0	0	SAU2EN	0	WUTEN	DFLEN

xxxEN	Input clock control
0	Stops input clock supply.
1	Supplies input clock.

Caution Be sure to clear the following bits to 0.
78K0R/HC3: Bit 2 of the PER0 register, bits 2 to 7 of the PER1 register
78K0R/HE3, 78K0R/HF3, 78K0R/HG3: Bits 2, 4 to 7 of the PER1 register

Remark

ADCEN:	Control of the A/D converter input clock
LIN1EN:	Control of the LIN-UART1 input clock
LIN0EN:	Control of the LIN-UART0 input clock
SAU1EN:	Control of the serial array unit 1 input clock
SAU0EN:	Control of the serial array unit 0 input clock
TAU2EN:	Control of the timer array unit 2 input clock
TAU1EN:	Control of the timer array unit 1 input clock
TAU0EN:	Control of the timer array unit 0 input clock
SAU2EN:	Control of the serial array unit 2 input clock
WUTEN:	Control of the 16-bit wakeup timer input clock
DFLEN:	Control of the data flash 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 18 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 the CLS that the CPU clock is operating on the PLL clock.
Confirm with the MCS that the main system clock is operating on a clock other than the high-speed system clock. When CLS = 0 and MCS = 1, the high-speed system clock (clock through mode ($f_{PLL} = f_X$)) or a PLL clock that is a multiple of the high-speed system clock is supplied to the CPU, so change the CPU clock to the internal low-speed oscillation clock or internal high-speed oscillation clock.

CLS	MCS	CPU Clock Status	
		Main System Clock	CPU/Peripheral Hardware Clock
0	0	Internal high-speed oscillation clock	PLL clock
0	1	High-speed system clock	PLL clock
1	×	−	Internal low-speed oscillation clock

Remark f_{PLL}: PLL clock frequency
f_X: X1 clock oscillation frequency

<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 OSTC 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.

5.7.2 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 and selecting the high-speed system clock as the clock for the main memory (MCM0 = 1)[Note]
(See **5.7.2 (1) Example of setting procedure when restarting oscillation of the internal high-speed oscillation clock**).

Note The setting of <1> is not necessary when the internal high-speed oscillation clock is operating.

<2> Setting PLL clock oscillation[Note]
(See **5.7.3 (1) Example of setting procedure when oscillating PLL clock** or **(2) Example of setting procedure when stopping PLL clock**.)

Note The setting of <2> is not necessary when the PLL clock is not used.

<3> Setting the PLL clock as 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_{PLL}
	0	0	1	$f_{PLL}/2$
	0	1	0	$f_{PLL}/2^2$
	0	1	1	$f_{PLL}/2^3$
	1	0	0	$f_{PLL}/2^4$
	1	0	1	$f_{PLL}/2^5$

Caution If switching the main system 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.

(a) When executing the STOP instruction
(b) When 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 18 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 the CLS that the CPU clock is operating on the PLL clock.

Confirm with the MCS that the main system clock 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 (clock through mode (f_{PLL} = f_{IH})) or a PLL clock that is a multiple of the internal high-speed oscillation clock is supplied to the CPU, so change the CPU clock to the internal low-speed oscillation clock or internal high-speed oscillation clock.

CLS	MCS	CPU Clock Status	
		Main System Clock	CPU/Peripheral Hardware Clock
0	0	Internal high-speed oscillation clock	PLL clock
0	1	High-speed system clock	PLL clock
1	×	−	Internal low-speed oscillation clock

Remark f_{PLL}: PLL clock frequency
f_{IH}: Internal high-speed oscillation clock

<2> Enabling CSC manipulation

Manipulating CSC is enabled by setting GDCSC to 1.

<3> 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.

5.7.3 Example of controlling subclock

An external clock input can be input as the subclock from the EXCLKS pin.

The subclock cannot be used as the CPU clock. The wakeup timer is the only hardware operating on the subclock.

The following describes examples of setting procedures for the following cases.

(1) When using an external subclock

(2) When stopping the subclock

(1) Example of setting procedure when using an external sub-clock

<1> Setting P124/EXCLKS pin (CMC register)

EXCLK	OSCSEL	EXCLKS	OSCSELS	0	0	0	AMPH
0/1	0/1	1	1	0	0	0	×

Remarks 1. ×: don't care

2. For setting of the P121/X1 and P122/X2 pins, see in **5.7.1 Example of controlling high-speed system clock for the settings**.

<2> Enabling CSC manipulation

Manipulating CSC is enabled by setting GDCSC to 1.

<3> Controlling external subclock input (CSC register)
When XTSTOP is cleared to 0, the input of the external subclock is enabled.

Cautions 1. The CMC register can be written only once by using an 8-bit memory manipulation instruction after reset release. Consequently, the EXCLK and OSCSEL bit values must be set at the same time. See 5.7.1 Example of controlling high-speed system clock for details of the EXCLK and OSCSEL bits.

2. Set the external subclock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 29 ELECTRICAL SPECIFICATIONS).

(2) Example of setting procedure when stopping the subclock

<1> Stopping the subclock (CSC register)
When XTSTOP is set to 1, the subclock is stopped.

Caution Stop the operation of peripheral hardware that is operating on the subclock when setting XTSTOP to 1.

5.7.4 Example of controlling PLL clock

The following describes examples of setting procedures for the following cases.

(1) When oscillating the PLL clock
(2) When stopping the PLL clock

(1) Example of setting procedure when oscillating PLL clock

<1> Enabling manipulation of PLLCTL
Manipulating PLLCTL is enabled by setting GDPLL to 1.

<2> Selecting PLL input clock (PLLCTL register)
Use PLLDIV0 to select the PLL input clock (f_{PLLI}).
When PLLDIV0 = 0, the f_{PLLI} is 4 MHz.
When PLLDIV0 = 1, the f_{PLLI} is 8 MHz.

<3> Selecting PLL output clock (PLLCTL register)
Use PLLDIV1 to select the PLL output clock (f_{PLLO}).

<4> Starting PLL operation
The PLL clock starts oscillating by setting PLLON to 1.

<5> Confirming the PLL status (PLLSTS register)
Use LOCK to confirm that the PLL is in a locked state (LOCK = 1).

<6> Selecting PLL clock mode (PLLCTL register)
Use SELPLL to select the PLL clock mode.
Set SELPLL = 1 and set the PLL clock select mode (a clock that is a multiple of f_{MAIN}[Note]).

Note The PLL multiplication number is set by using bit 2 (OPTPLL) of the option byte (000C1H). The combinations of the PLL input clocks and multiplication numbers are predetermined. See **5.3 (11) PLL control register (PLLCTL)** for details.

Remark f_{PLLI}: PLL input clock frequency, f_{MAIN}: Main system clock frequency

<7> Confirming PLL clock mode (PLLSTS register)
Use SELPLLS to confirm that PLL clock mode is selected (SELPLLS = 1).

<8> Disabling manipulation of PLLCTL
Manipulating PLLCTL is disabled by clearing GDPLL to 0.

(2) Example of setting procedure when stopping PLL clock

The PLL clock can be stopped by using the following two methods.

(a) When executing a STOP instruction and transitioning to STOP mode

(b) When setting PLLON to 0 and stopping the PLL clock

(a) When executing a STOP instruction

<1> Setting of peripheral hardware
Stop all peripheral hardware that cannot be used in STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 18 STANDBY FUNCTION**.)

<2> Setting the X1 clock oscillation stabilization time after standby release
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 a STOP instruction
When a STOP instruction is executed, the system is placed in the STOP mode and the PLL clock is stopped.

(b) When setting PLLON to 0 and stopping the PLL clock

<1> Enabling manipulation of PLLCTL
Manipulating PLLCTL is enabled by setting GDPLL to 1.

<2> Selecting PLL clock mode (PLLCTL register)
Use SELPLL to select the PLL clock mode.
Clear SELPLL to 0 to set to clock through mode ($f_{PLL} = f_{MAIN}$).

Remark f_{PLL}: PLL clock frequency, f_{MAIN}: Main system clock frequency

<3> Checking PLL clock mode (PLLSTS register)
Check that the clock through mode is set (SELPLLS = 0).

<4> Stopping PLL operation
PLL clock oscillation is stopped by clearing PLLON to 0.

<5> Disabling manipulation of PLLCTL
Manipulating PLLCTL is disabled by clearing GDPLL to 0.

5.7.5 Example of controlling internal low-speed oscillation clock

The internal low-speed oscillator is incorporated in the 78K0R/Hx3 (30 kHz (TYP.)).

Oscillation operation of the internal low-speed oscillator after a reset release can be controlled by using bit 7 (LIOUSE) of the option byte (000C1H).

The internal low-speed oscillator starts to oscillate automatically, when its operation has been enabled by using bit 7 (LIOUSE) of the option byte (000C1H) after a reset release.

The following describes examples of setting procedures for the following cases.

(1) When oscillating the internal low-speed oscillation clock

(2) When using the internal low-speed oscillation clock as the CPU clock

(1) Example of setting procedure when oscillating the internal low-speed oscillation clock

<1> Enabling operation of internal low-speed oscillation (option byte (000C1H))
Operation is enabled by setting LIOUSE to “1”.

LIOUSE	LIOSTOPB	LIOSYSB	CLKMB	RESOUTB	OPTPLL	SEL4M	LVIOFF
1	×	×	×	×	×	×	×

Remarks 1. See **CHAPTER 23 OPTION BYTE** for details of setting the option byte.
2. ×: don’t care

<2> Starting operation of internal low-speed oscillation
Oscillation is started automatically after reset release.

(2) Example of setting procedure when using the internal low-speed oscillation clock as the CPU clock

<1> Enabling operation of internal low-speed oscillation and selection of CPU/peripheral hardware clock (option byte (000C1H))
Operation and selection of CPU clock are enabled by setting LIOUSE to “1” and LIOSYSB to “0”.

LIOUSE	LIOSTOPB	LIOSYSB	CLKMB	RESOUTB	OPTPLL	SEL4M	LVIOFF
1	×	0	×	×	×	×	×

Remarks 1. See **CHAPTER 23 OPTION BYTE** for details of setting the option byte.
2. ×: don’t care

<2> Setting the internal low-speed oscillation clock as the CPU/peripheral hardware clock (CKC register)

CLS	CSS	MCS	MCM0	0	MDIV2	MDIV1	MDIV0
1	1	×	×	0	×	×	×

Remark ×: don’t care

Caution **When the internal low-speed oscillation clock is used as the CPU clock, the internal low-speed oscillation clock is also supplied to the peripheral hardware clock. At this time, the operation of the A/D converter cannot be guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapter describing each peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS.**

Table 5-4. Control of Operating/Stopping Internal Low-Speed Oscillation and Watchdog Timer

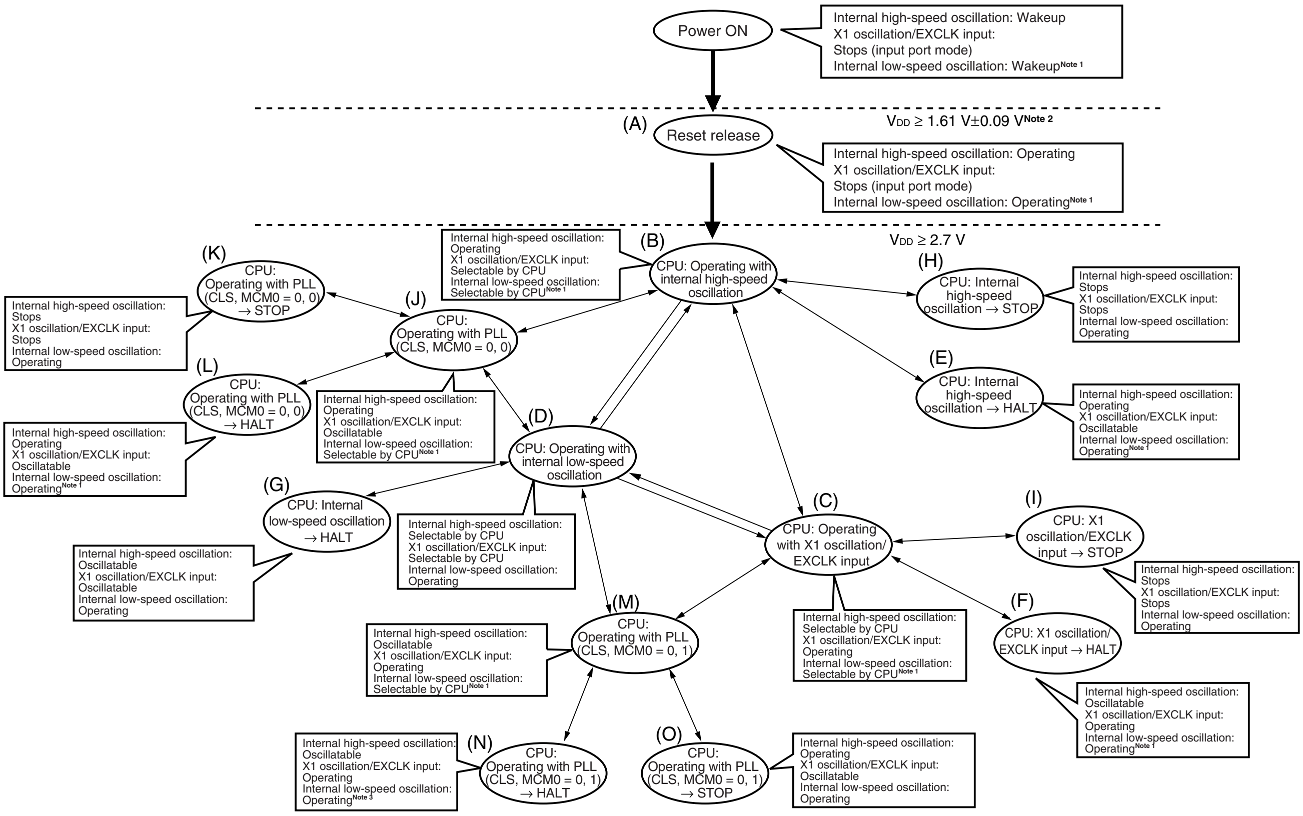
Option Byte (000C1H)		Option Byte (000C0H)		CPU Status	Internal Low-Speed Oscillation Clock Operating State	Watchdog Timer Operating State
LIOUSE	LIOSTOPB	WDTON	WDSTBYON			
0	×	×	×	×	Stops	Stops
1	0	0	×	RUN/HALT	Operating	Stops
				STOP	Stops	Stops
1	0	1	0	RUN	Operating	Operating
				HALT	Operating	Stops
				STOP	Stops	Stops
1	0	1	1	RUN/HALT	Operating	Operating
				STOP	Stops	Stops
1	1	0	×	×	Operating	Stops
1	1	1	0	RUN	Operating	Operating
				HALT	Operating	Stops
				STOP	Operating	Stops
1	1	1	1	×	Operating	Operating

Caution Executing a STOP instruction is prohibited when the internal low-speed oscillation clock is selected as the CPU clock.

5.7.6 CPU clock status transition diagram

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

Figure 5-18. CPU Clock Status Transition Diagram


Power ON
Internal high-speed oscillation: Wakeup
X1 oscillation/EXCLK input:
Stops (input port mode)
Internal low-speed oscillation: Wakeup Note 1
(A)
Reset release
VDD ≥ 1.61 V±0.09 V Note 2
Internal high-speed oscillation: Operating
X1 oscillation/EXCLK input:
Stops (input port mode)
Internal low-speed oscillation: Operating Note 1
VDD ≥ 2.7 V
(B)
CPU: Operating with internal high-speed oscillation
Internal high-speed oscillation: Operating
X1 oscillation/EXCLK input: Selectable by CPU
Internal low-speed oscillation: Selectable by CPU Note 1
(K)
CPU: Operating with PLL (CLS, MCM0 = 0, 0) → STOP
Internal high-speed oscillation: Stops
X1 oscillation/EXCLK input: Stops
Internal low-speed oscillation: Operating
(J)
CPU: Operating with PLL (CLS, MCM0 = 0, 0)
Internal high-speed oscillation: Operating
X1 oscillation/EXCLK input: Oscillatable
Internal low-speed oscillation: Selectable by CPU Note 1
(L)
CPU: Operating with PLL (CLS, MCM0 = 0, 0) → HALT
Internal high-speed oscillation: Operating
X1 oscillation/EXCLK input: Oscillatable
Internal low-speed oscillation: Operating Note 1
(H)
CPU: Internal high-speed oscillation → STOP
Internal high-speed oscillation: Stops
X1 oscillation/EXCLK input: Stops
Internal low-speed oscillation: Operating
(E)
CPU: Internal high-speed oscillation → HALT
Internal high-speed oscillation: Operating
X1 oscillation/EXCLK input: Oscillatable
Internal low-speed oscillation: Operating Note 1
(D)
CPU: Operating with internal low-speed oscillation
Internal high-speed oscillation: Selectable by CPU
X1 oscillation/EXCLK input: Selectable by CPU
Internal low-speed oscillation: Operating
(G)
CPU: Internal low-speed oscillation → HALT
Internal high-speed oscillation: Oscillatable
X1 oscillation/EXCLK input: Oscillatable
Internal low-speed oscillation: Operating
(C)
CPU: Operating with X1 oscillation/ EXCLK input
Internal high-speed oscillation: Selectable by CPU
X1 oscillation/EXCLK input: Operating
Internal low-speed oscillation: Selectable by CPU Note 1
(I)
CPU: X1 oscillation/EXCLK input → STOP
Internal high-speed oscillation: Stops
X1 oscillation/EXCLK input: Stops
Internal low-speed oscillation: Operating
(F)
CPU: X1 oscillation/ EXCLK input → HALT
Internal high-speed oscillation: Oscillatable
X1 oscillation/EXCLK input: Operating
Internal low-speed oscillation: Operating Note 1
(M)
CPU: Operating with PLL (CLS, MCM0 = 0, 1)
Internal high-speed oscillation: Oscillatable
X1 oscillation/EXCLK input: Operating
Internal low-speed oscillation: Selectable by CPU Note 1
(N)
CPU: Operating with PLL (CLS, MCM0 = 0, 1) → HALT
Internal high-speed oscillation: Oscillatable
X1 oscillation/EXCLK input: Operating
Internal low-speed oscillation: Operating Note 3
(O)
CPU: Operating with PLL (CLS, MCM0 = 0, 1) → STOP
Internal high-speed oscillation: Operating
X1 oscillation/EXCLK input: Oscillatable
Internal low-speed oscillation: Operating

Notes 1. When LIOUSE = 1 and LIOSYSB = 0. When LIOUSE = 0, internal low-speed oscillation stops.

2. When LIOUSE = 1. When LIOUSE = 0, or LIOUSE = 1 and LIOSTOPB = 0, internal low-speed oscillation stops.

Caution Transition in the order of (B), (D), (C), or (C), (D), (B) is prohibited.

Remark If the low-voltage 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 2.93 V ±0.2 V.

After the reset operation, the status will shift to (B) in the above figure.

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

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

(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) →

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

<R> **Note** 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 29 ELECTRICAL SPECIFICATIONS).

Remark ×: don't care

(3) CPU operating with internal low-speed oscillation clock (D) after reset release (A)
(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence) →

Setting	Option Byte (000C1H)		CKC Register
Status Transition	LIOUSE	LIOSYSB	CSS
(A) → (B) → (D)	1	0	1

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

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

(4) CPU clock changing from internal high-speed oscillation clock (B) to high-speed system clock (C)

(Setting sequence) →

Setting Flag	CMC Register[Note 1]			OSTS Register	CSC Register	OSMC Register	OSTC Register	CKC Register
Status Transition	EXCLK	OSCSEL	AMPH		MSTOP	FSEL		MCM0
(B) → (C) (X1 clock: 2 MHz ≤ f_X ≤ 10 MHz)	0	1	0	**Note 2**	0	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < f_X ≤ 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 of OSTS 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} ≤ 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 29 ELECTRICAL SPECIFICATIONS).

Remark ×: don't care

(5) CPU clock changing from internal high-speed oscillation clock (B) or high-speed system clock (C) to internal low-speed oscillation clock (D)

(Setting sequence) →

Setting	Option Byte (000C1H)		CKC Register
Status Transition	LIOUSE	LIOSYSB	CSS
(B) → (D) (C) → (D)	1	0	1

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

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

(6)
- **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 internal low-speed oscillation clock (D)**

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

(7)
- **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)	In X1 stop	Stopping peripheral functions that cannot operate in STOP mode	–	Executing STOP instruction
	In X1 oscillation		Sets the OSTS register	
(C) → (I)				

(8)
- **CPU transitioning from internal high-speed oscillation clock (B) to PLL clock (J) by internal high-speed oscillation clock**
- **CPU transitioning from high-speed system clock (C) to PLL clock (M) by high-speed system clock**

(Setting sequence) →

Setting Flag Status Transition	Option Byte (000C1H)	PLLSTL Register			PLLSTS Register	PLLCTL Register	PLLSTS Register
	OPTPLL	PLLDIV0	PLLDIV1	PLLON	LOCK	SELPLL	SELPLLS
(B) → (J) (C) → (M)	0/1	0/1	0/1	1	Must be checked	1	Must be checked

(9) CPU clock changing from high-speed system clock (C) to internal high-speed oscillation clock (B)

(Setting sequence) →

Setting Flag 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 (O) in Table 5-5 correspond to (A) to (O) in Figure 5-18.

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

(10) CPU clock changing from internal low-speed oscillation clock (D) to internal high-speed oscillation clock (B)

(Setting sequence) →

Setting Flag Status Transition	CSC Register	CKC Register	
	HIOSTOP	MCM0	CSS
(D) → (B)	0	0	0

HIOSTOP: Unnecessary if the CPU is operating with the internal high-speed oscillation clock

MCM0: Changing prohibited

(11) CPU clock changing from internal low-speed oscillation clock (D) to high-speed system clock (C)

(Setting sequence) →

Setting Flag Status Transition	CMC Register[Note 1]			OSTS Register	CSC Register	OSMC Register	OSTC Register	CKC Register	
	EXCLK	OSCSEL	AMPH		MSTOP	FSEL		MCM0	CSS
(D) → (C) (X1 clock: 2 MHz ≤ f_X ≤ 10 MHz)	0	1	0	**Note 2**	0	0	Must be checked	1	0
(D) → (C) (X1 clock: 10 MHz < f_X ≤ 24 MHz)	0	1	1	**Note 2**	0	1 [Note 3]	Must be checked	1	0
(D) → (C) (external main clock)	1	1	0/1	**Note 2**	0	0/1	Must not be checked	1	0

EXCLK to AMPH: Unnecessary if these registers are already set

OSTS to OSTC: Unnecessary if the CPU is operating with the high-speed system clock

MCM0: Changing prohibited

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 of OSTS 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 $f_{CLK} \le 10$ 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 29 ELECTRICAL SPECIFICATIONS).

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

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

(12) • CPU transitioning from PLL clock (J) by internal high-speed oscillation clock to internal high-speed oscillation clock (B)

• CPU transitioning from PLL clock (M) by high-speed system clock to high-speed system clock (C)

(Setting sequence) →

Setting Flag	PLLCTL Register	PLLSTS Register	PLLCTL Register
Status Transition	SELPLL	SELPLLS	PLLON
(J) → (B) (M) → (C)	0	Must be checked	0

(13) • CPU transitioning from PLL clock (J) by internal high-speed oscillation clock to internal low-speed oscillation clock (D)

• CPU transitioning from PLL clock (M) by high-speed system clock to internal low-speed oscillation clock (D)

(Setting sequence) →

Setting	Option Byte (000C1H)		CKC Register
Status Transition	LIOUSE	LIOSYSB	CSS
(J) → (D) (M) → (D)	1	0	1

(14) • STOP mode (K) set while CPU is operating with PLL clock (J) by internal high-speed oscillation clock

• STOP mode (O) set while CPU is operating with PLL clock (M) by high-speed system clock

(Setting sequence) →

Status Transition		Setting		
(J) → (K)		Stopping peripheral functions that cannot operate in STOP mode	–	Executing STOP instruction
(M) → (O)	In X1 oscillation		Sets the OSTS register	
	External clock		–	

(15) • HALT mode (L) set while CPU is operating with PLL clock (J) by internal high-speed oscillation clock

• HALT mode (N) set while CPU is operating with PLL clock (M) by high-speed system clock

Status Transition	Setting
(J) → (L) (M) → (N)	Executing HALT instruction

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

5.7.7 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 5-6. Changing CPU Clock

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	
	Internal low-speed oscillation clock	Stabilization of internal low-speed oscillation • CSS = 1, LIOUSE = 1, LIOSYSB = 0	
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.)	–
	Internal low-speed oscillation clock	Stabilization of internal low-speed oscillation • CSS = 1, LIOUSE = 1, LIOSYSB = 0	X1 oscillation can be stopped (MSTOP = 1).
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.)	–
	Internal low-speed oscillation clock	Stabilization of internal low-speed oscillation • CSS = 1, LIOUSE = 1, LIOSYSB = 0	External main system clock input can be disabled (MSTOP = 1).
Internal low-speed oscillation 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	–
	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	

Remark LIOUSE: Bit 7 of the option byte (000C1H)
LIOSYSB: Bit 5 of the option byte (000C1H)

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

By setting bits 0 to 2 and 4 (MDIV0 to MDIV2, MCM0) of the system clock control register (CKC), the 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 5-7** to **Table 5-11**).

Whether the main system clock, which is to be the CPU 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.

Table 5-7. Maximum Time Required for Main System Clock Switchover

Clock A	Switching Directions	Clock B	Type
f_{IH}	←→	f_{MX}	Type 2 (see **Table 5-9**)
f_{PLL}	←→	f_{IL}	Type 3 (see **Table 5-10**)
f_{PLL}	←→ (Changing the division ratio)	f_{PLL}	Type 1 (see **Table 5-8**)
f_{IL}	←→ (Changing the division ratio)	f_{IL}	Type 1 (see **Table 5-8**)
f_{PLL} (When clock through mode (f_{MAIN}) is selected using SELPLL)	←→ (Changing the PLL operation mode)	f_{PLL} (When PLL clock select mode (f_{PLLO}) is selected using SELPLL)	Type 4 (see **Table 5-11**)

Table 5-8. Maximum Number of Clocks Required in Type 1

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 5-9. Maximum Number of Clocks Required in Type 2

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 5-10. Maximum Number of Clocks Required in Type 3

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

Table 5-11. Maximum Number of Clocks Required in Type 4

Set Value Before Switchover	Set Value After Switchover	
SELPLL	SELPLL	
	0 (f_{PLL} = f_{MAIN})	1 (f_{PLL} = f_{PLLO})
0 (f_{PLL} = f_{MAIN})		1 + f_{MAIN}/f_{PLL} clock
1 (f_{PLL} = f_{PLLO})	$2f_{PLL}/f_{MAIN}$ clock	

Remarks 1. The number of clocks listed in Table 5-8 to Table 5-11 is the number of CPU clocks before switchover.

2. Calculate the number of clocks in Table 5-8 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)

$1 + f_{IH}/f_{MX} = 1 + 8/10 = 1 + 0.8 = 1.8 \rightarrow 2$ clocks

5.7.9 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	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock)	MSTOP = 1
External main system clock		
PLL clock	SELPLLS = 0 or CLS = 1 (The CPU is operating on a clock other than the PLL clock)	PLLON = 0

CHAPTER 6 TIMER ARRAY UNIT

TAU \ Product	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
16-bit timer	16	20		24
Timer array unit	2	3		

The timer array unit has up to three units. The timer array unit 0 has eight 16-bit timers and the timer array units 1 and 2 have four or eight 16-bit timers. Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more "channels" can be used to create a high-accuracy timer.

Single-operation Function	Combination-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	• PWM output • One-shot pulse output • Multiple PWM output

Channels 2 and 3 of the timer array unit 1 can be used to realize LIN-bus reception processing in combination with LIN-UART0, 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

Single-operation functions are those functions that can be used for any channel regardless of the operation mode of the other channel (for details, see **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 INTTMmn is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOmn).

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(3) External event counter

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

(4) Divider function

A clock input from a timer input pin (TImn) is divided and output from an output pin (TOmn).

(5) Input pulse interval measurement

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

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

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

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3
[When frequency divider is used]
m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 7)
mn = 05, 11: 78K0R/HC3, 78K0R/HE3, mn = 00 to 02, 04 to 07, 10 to 17: 78K0R/HF3,
mn = 00 to 07, 10 to 17: 78K0R/HG3

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, see **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.

(3) Multiple PWM (Pulse Width Modulator) output

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

6.1.3 LIN-bus supporting function (channels 2 and 3 of the timer array unit 1 only)

(1) Detection of wakeup signal

The timer starts counting at the falling edge of a signal input to the serial data input pin (LRxD0, LRxD1) of LIN-UART 0, 1 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 (LRxD0, LRxD1) of LIN-UART0, 1 after a wakeup signal is detected, and the count value of the timer is captured at the rising edge. In this way, a lowlevel 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 (LRxD0, LRxD1) of LIN-UART0, 1 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.

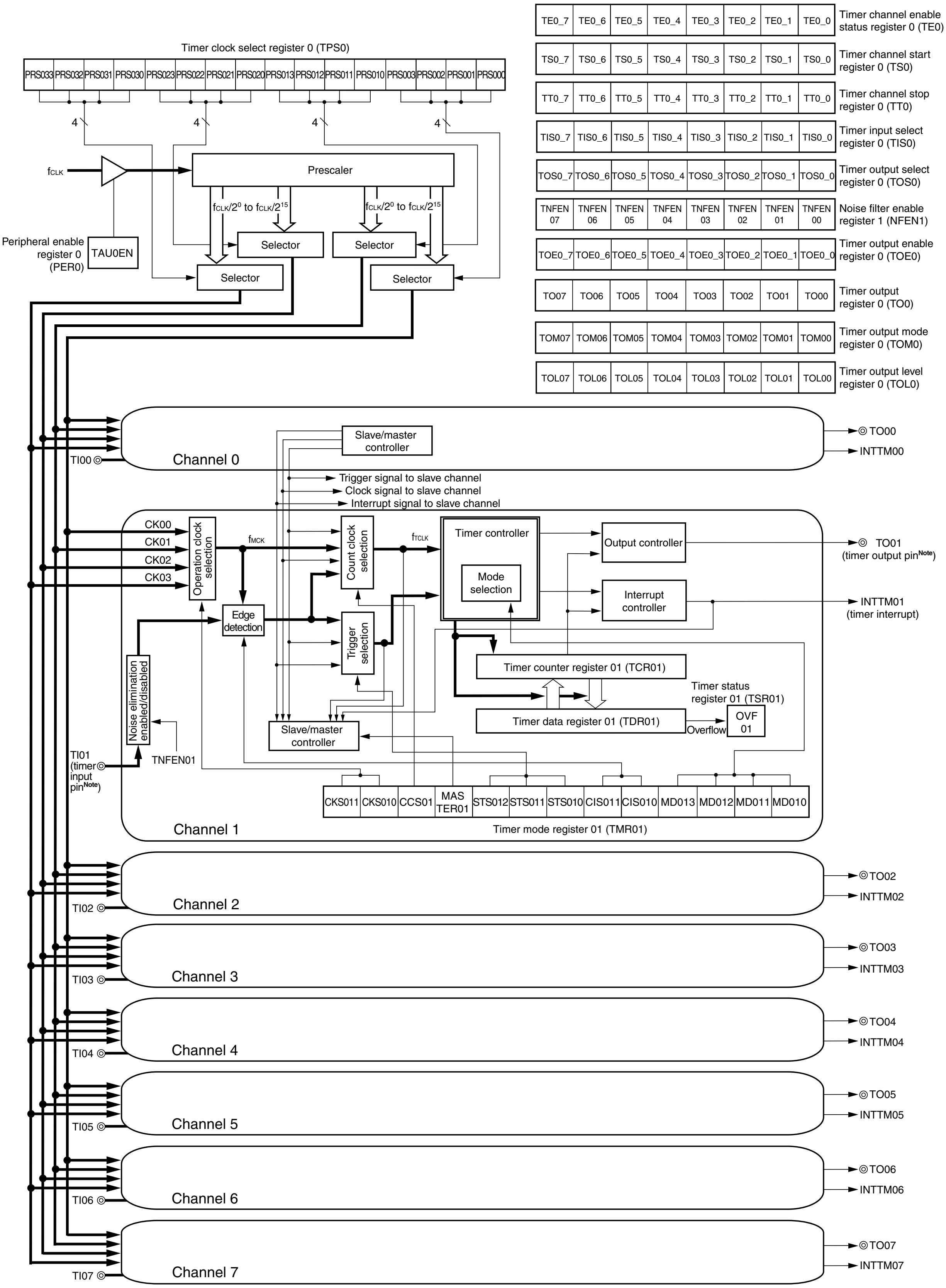
Table 6-1. Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer counter register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI07, TI10 to TI17, TI20 to TI27 pins
Timer output	TO00 to TO07, TO10 to TO17, TO20 to TO27 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 output enable register m (TOEm) • Timer output register m (TOm) • Timer output level register m (TOLm) • Timer output mode register m (TOMm)
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register mn (TSRmn) • Noise filter enable registers 1 to 3 (NFEN1 to NFEN3) • Timer input select registers 0, 1 (TIS0, TIS1) • Timer output select registers 0, 1 (TOS0, TOS1) • Serial communication pin select register (STSEL) • Port mode registers 0, 1, 3 to 7, 12, 15 (PM0, PM1, PM3 to PM7, PM12, PM15) • Port registers 0, 1, 3 to 7, 12, 15 (P0, P1, P3 to P7, P12, P15)

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

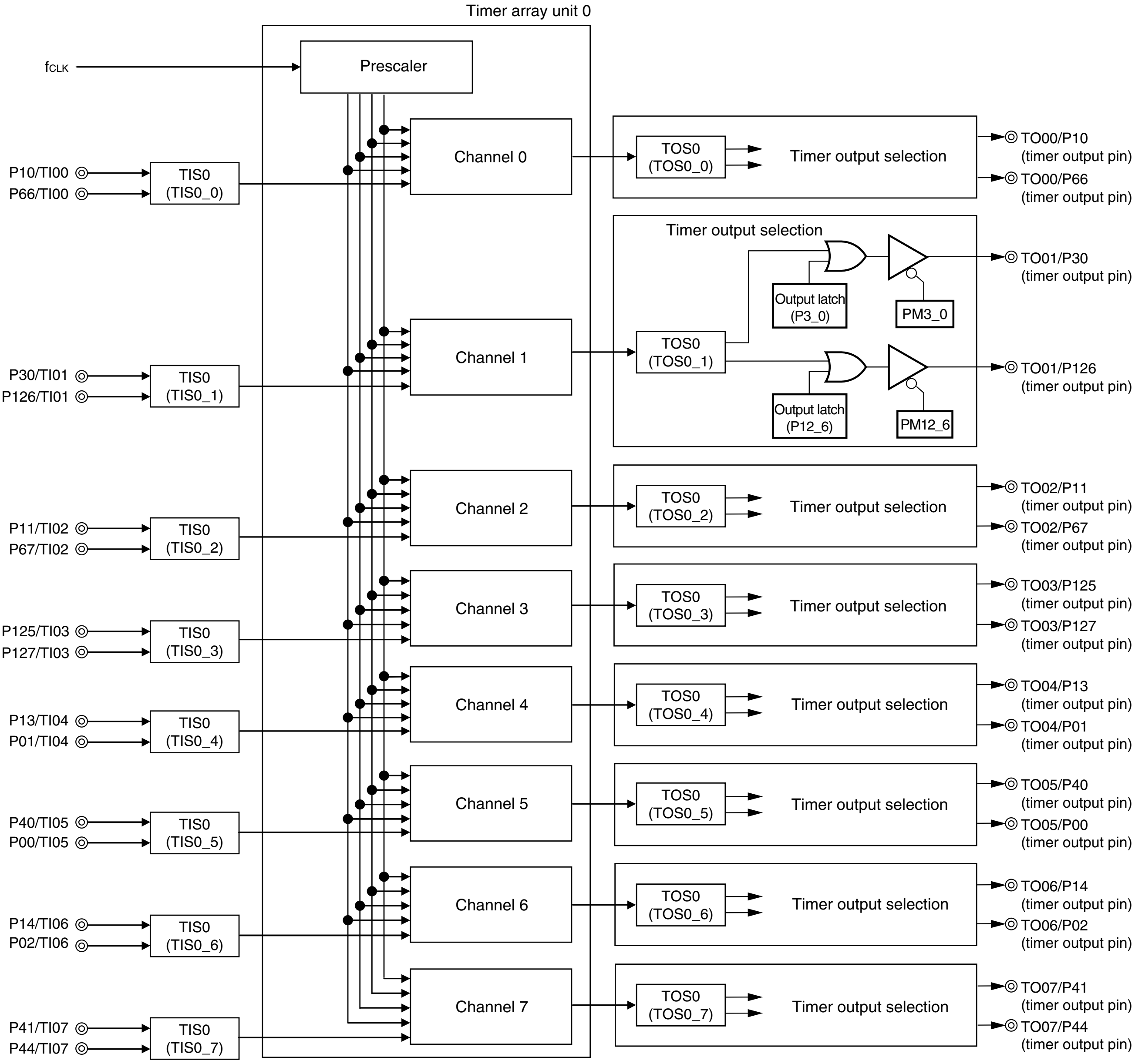
Figure 6-1, Figure 6-3, and Figure 6-5 show the block diagrams.

Figure 6-1. Block Diagram of Timer Array Unit 0


TE0_7 TE0_6 TE0_5 TE0_4 TE0_3 TE0_2 TE0_1 TE0_0
Timer channel enable status register 0 (TE0)
TS0_7 TS0_6 TS0_5 TS0_4 TS0_3 TS0_2 TS0_1 TS0_0
Timer channel start register 0 (TS0)
TT0_7 TT0_6 TT0_5 TT0_4 TT0_3 TT0_2 TT0_1 TT0_0
Timer channel stop register 0 (TT0)
TIS0_7 TIS0_6 TIS0_5 TIS0_4 TIS0_3 TIS0_2 TIS0_1 TIS0_0
Timer input select register 0 (TIS0)
TOS0_7 TOS0_6 TOS0_5 TOS0_4 TOS0_3 TOS0_2 TOS0_1 TOS0_0
Timer output select register 0 (TOS0)
TNFEN07 TNFEN06 TNFEN05 TNFEN04 TNFEN03 TNFEN02 TNFEN01 TNFEN00
Noise filter enable register 1 (NFEN1)
TOE0_7 TOE0_6 TOE0_5 TOE0_4 TOE0_3 TOE0_2 TOE0_1 TOE0_0
Timer output enable register 0 (TOE0)
TO07 TO06 TO05 TO04 TO03 TO02 TO01 TO00
Timer output register 0 (TO0)
TOM07 TOM06 TOM05 TOM04 TOM03 TOM02 TOM01 TOM00
Timer output mode register 0 (TOM0)
TOL07 TOL06 TOL05 TOL04 TOL03 TOL02 TOL01 TOL00
Timer output level register 0 (TOL0)
Timer clock select register 0 (TPS0)
PRS033 PRS032 PRS031 PRS030 PRS023 PRS022 PRS021 PRS020 PRS013 PRS012 PRS011 PRS010 PRS003 PRS002 PRS001 PRS000
4
4
4
4
fCLK
Prescaler
fCLK/2^0 to fCLK/2^15
fCLK/2^0 to fCLK/2^15
Peripheral enable register 0 (PER0)
TAU0EN
Selector
Selector
Selector
Selector
Slave/master controller
Channel 0
TI00
TO00
INTTM00
Trigger signal to slave channel
Clock signal to slave channel
Interrupt signal to slave channel
CK00
CK01
CK02
CK03
Operation clock selection
fMCK
Count clock selection
fTCLK
Timer controller
Mode selection
Output controller
Interrupt controller
TO01 (timer output pin^Note)
INTTM01 (timer interrupt)
Edge detection
Trigger selection
Noise elimination enabled/disabled
TNFEN01
TI01 (timer input pin^Note)
Timer counter register 01 (TCR01)
Timer status register 01 (TSR01)
Timer data register 01 (TDR01)
Overflow
OVF 01
Slave/master controller
CKS011 CKS010 CCS01 MASTER01 STS012 STS011 STS010 CIS011 CIS010 MD013 MD012 MD011 MD010
Timer mode register 01 (TMR01)
Channel 1
Channel 2
TI02
TO02
INTTM02
Channel 3
TI03
TO03
INTTM03
Channel 4
TI04
TO04
INTTM04
Channel 5
TI05
TO05
INTTM05
Channel 6
TI06
TO06
INTTM06
Channel 7
TI07
TO07
INTTM07

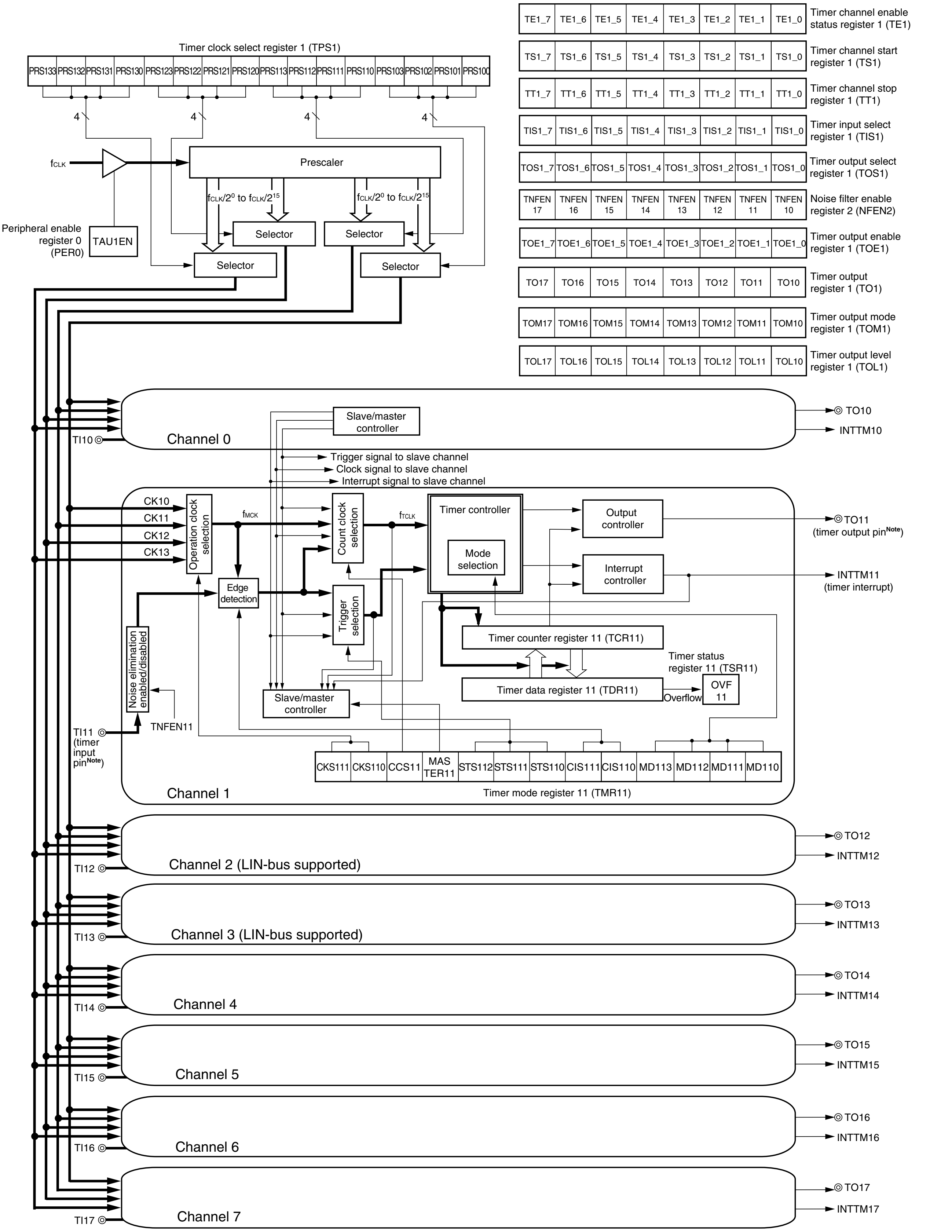
Note See **Figure 6-2** for timer input pin selection and timer output pin selection.

Figure 6-2. Port Configuration Diagram of Timer Array Unit 0



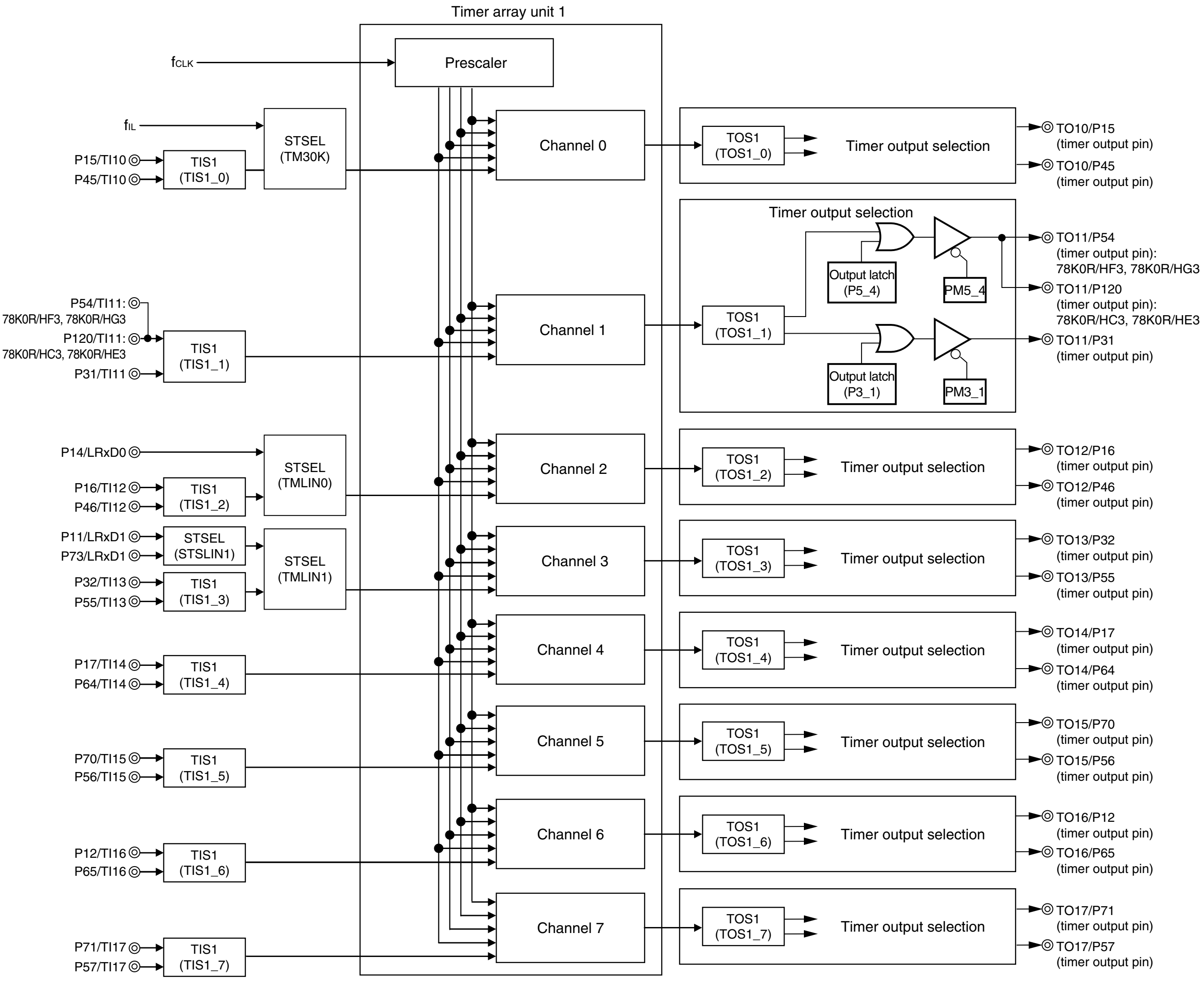
Remark The pins mounted differ depending on the product. See **6.3 (13) Timer input select registers 0, 1 (TIS0, TIS1)** and **6.3 (14) Timer output select registers 0, 1 (TOS0, TOS1)** for details.

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


TE1_7 TE1_6 TE1_5 TE1_4 TE1_3 TE1_2 TE1_1 TE1_0
Timer channel enable status register 1 (TE1)
TS1_7 TS1_6 TS1_5 TS1_4 TS1_3 TS1_2 TS1_1 TS1_0
Timer channel start register 1 (TS1)
TT1_7 TT1_6 TT1_5 TT1_4 TT1_3 TT1_2 TT1_1 TT1_0
Timer channel stop register 1 (TT1)
TIS1_7 TIS1_6 TIS1_5 TIS1_4 TIS1_3 TIS1_2 TIS1_1 TIS1_0
Timer input select register 1 (TIS1)
TOS1_7 TOS1_6 TOS1_5 TOS1_4 TOS1_3 TOS1_2 TOS1_1 TOS1_0
Timer output select register 1 (TOS1)
TNFEN17 TNFEN16 TNFEN15 TNFEN14 TNFEN13 TNFEN12 TNFEN11 TNFEN10
Noise filter enable register 2 (NFEN2)
TOE1_7 TOE1_6 TOE1_5 TOE1_4 TOE1_3 TOE1_2 TOE1_1 TOE1_0
Timer output enable register 1 (TOE1)
TO17 TO16 TO15 TO14 TO13 TO12 TO11 TO10
Timer output register 1 (TO1)
TOM17 TOM16 TOM15 TOM14 TOM13 TOM12 TOM11 TOM10
Timer output mode register 1 (TOM1)
TOL17 TOL16 TOL15 TOL14 TOL13 TOL12 TOL11 TOL10
Timer output level register 1 (TOL1)
Timer clock select register 1 (TPS1)
PRS133 PRS132 PRS131 PRS130 PRS123 PRS122 PRS121 PRS120 PRS113 PRS112 PRS111 PRS110 PRS103 PRS102 PRS101 PRS100
4
4
4
4
fCLK
Prescaler
fCLK/2^0 to fCLK/2^15
fCLK/2^0 to fCLK/2^15
Peripheral enable register 0 (PER0)
TAU1EN
Selector
Selector
Selector
Selector
Slave/master controller
Channel 0
TI10
TO10
INTTM10
Trigger signal to slave channel
Clock signal to slave channel
Interrupt signal to slave channel
CK10
CK11
CK12
CK13
Operation clock selection
fMCK
Count clock selection
fTCLK
Timer controller
Mode selection
Output controller
Interrupt controller
TO11 (timer output pin^Note)
INTTM11 (timer interrupt)
Edge detection
Trigger selection
Noise elimination enabled/disabled
TNFEN11
TI11 (timer input pin^Note)
Timer counter register 11 (TCR11)
Timer status register 11 (TSR11)
Timer data register 11 (TDR11)
Overflow
OVF 11
Slave/master controller
CKS111 CKS110 CCS11 MASTER11 STS112 STS111 STS110 CIS111 CIS110 MD113 MD112 MD111 MD110
Timer mode register 11 (TMR11)
Channel 1
Channel 2 (LIN-bus supported)
TI12
TO12
INTTM12
Channel 3 (LIN-bus supported)
TI13
TO13
INTTM13
Channel 4
TI14
TO14
INTTM14
Channel 5
TI15
TO15
INTTM15
Channel 6
TI16
TO16
INTTM16
Channel 7
TI17
TO17
INTTM17

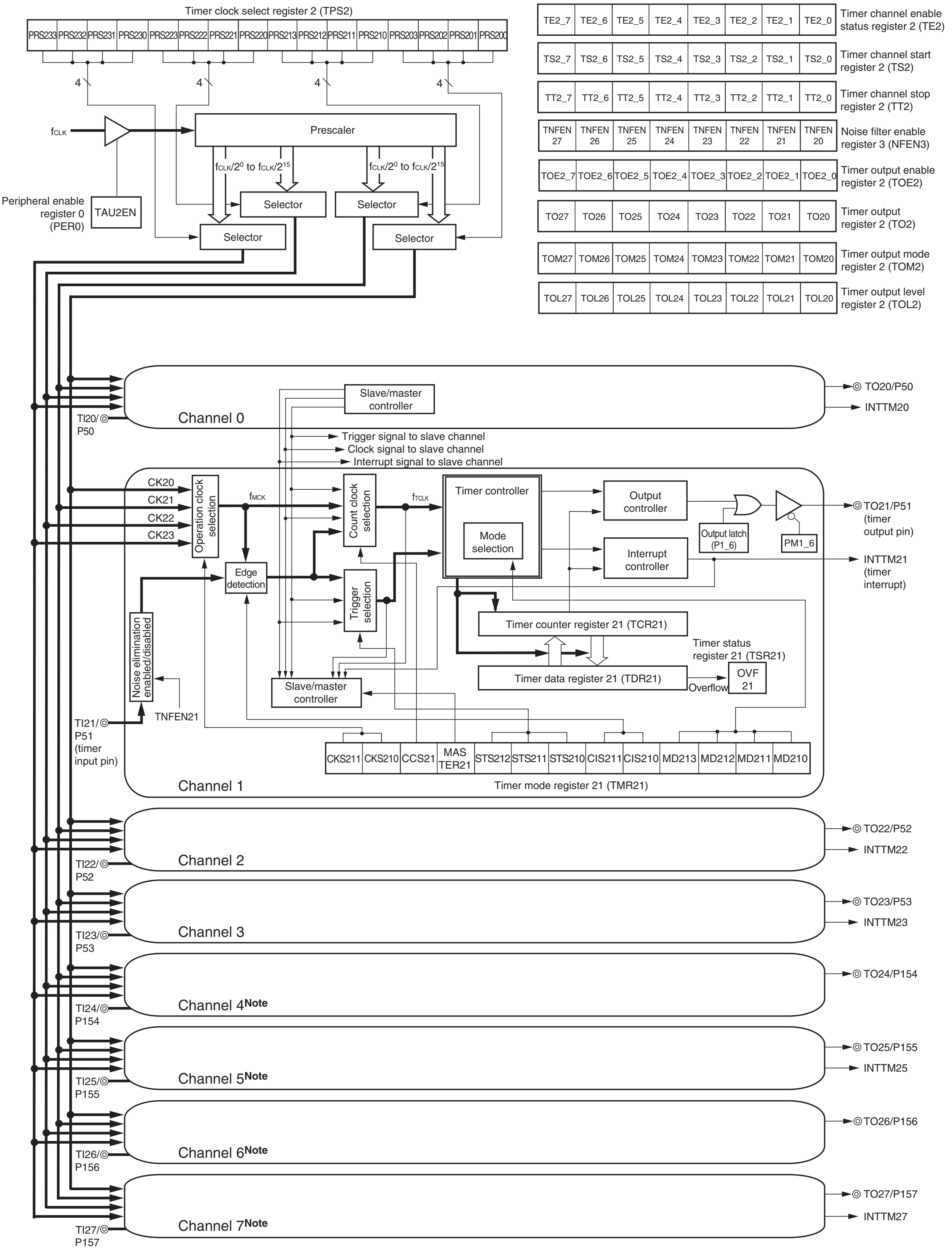
Note See **Figure 6-4** for timer input pin selection and timer output pin selection.

Figure 6-4. Port Configuration Diagram of Timer Array Unit 1



Remark The pins mounted differ depending on the product. See **6.3 (13) Timer input select registers 0, 1 (TIS0, TIS1)** and **6.3 (14) Timer output select registers 0, 1 (TOS0, TOS1)** for details.

Figure 6-5. Block Diagram of Timer Array Unit 2 (78K0R/HE3, 78K0R/HF3, 78K0R/HG3)


Timer clock select register 2 (TPS2)
PRS233 PRS232 PRS231 PRS230 PRS223 PRS222 PRS221 PRS220 PRS213 PRS212 PRS211 PRS210 PRS203 PRS202 PRS201 PRS200
4
4
4
4
fCLK
Prescaler
fCLK/2^0 to fCLK/2^15
fCLK/2^0 to fCLK/2^15
Peripheral enable register 0 (PER0)
TAU2EN
Selector
Selector
Selector
Selector
TE2_7 TE2_6 TE2_5 TE2_4 TE2_3 TE2_2 TE2_1 TE2_0
Timer channel enable status register 2 (TE2)
TS2_7 TS2_6 TS2_5 TS2_4 TS2_3 TS2_2 TS2_1 TS2_0
Timer channel start register 2 (TS2)
TT2_7 TT2_6 TT2_5 TT2_4 TT2_3 TT2_2 TT2_1 TT2_0
Timer channel stop register 2 (TT2)
TNFEN27 TNFEN26 TNFEN25 TNFEN24 TNFEN23 TNFEN22 TNFEN21 TNFEN20
Noise filter enable register 3 (NFEN3)
TOE2_7 TOE2_6 TOE2_5 TOE2_4 TOE2_3 TOE2_2 TOE2_1 TOE2_0
Timer output enable register 2 (TOE2)
TO27 TO26 TO25 TO24 TO23 TO22 TO21 TO20
Timer output register 2 (TO2)
TOM27 TOM26 TOM25 TOM24 TOM23 TOM22 TOM21 TOM20
Timer output mode register 2 (TOM2)
TOL27 TOL26 TOL25 TOL24 TOL23 TOL22 TOL21 TOL20
Timer output level register 2 (TOL2)
Slave/master controller
TI20/◎ P50
Channel 0
◎ TO20/P50
INTTM20
Trigger signal to slave channel
Clock signal to slave channel
Interrupt signal to slave channel
CK20
CK21
CK22
CK23
Operation clock selection
fMCK
Count clock selection
fTCLK
Timer controller
Mode selection
Output controller
Interrupt controller
Output latch (P1_6)
PM1_6
◎ TO21/P51 (timer output pin)
INTTM21 (timer interrupt)
Edge detection
Trigger selection
Noise elimination enabled/disabled
TNFEN21
Timer counter register 21 (TCR21)
Timer status register 21 (TSR21)
Timer data register 21 (TDR21)
Overflow
OVF 21
Slave/master controller
TI21/◎ P51 (timer input pin)
CKS211 CKS210 CCS21 MASTER21 STS212 STS211 STS210 CIS211 CIS210 MD213 MD212 MD211 MD210
Timer mode register 21 (TMR21)
Channel 1
Channel 2
TI22/◎ P52
◎ TO22/P52
INTTM22
Channel 3
TI23/◎ P53
◎ TO23/P53
INTTM23
Channel 4Note
TI24/◎ P154
◎ TO24/P154
Channel 5Note
TI25/◎ P155
◎ TO25/P155
INTTM25
Channel 6Note
TI26/◎ P156
◎ TO26/P156
Channel 7Note
TI27/◎ P157
◎ TO27/P157
INTTM27

Note 78K0R/HG3 only

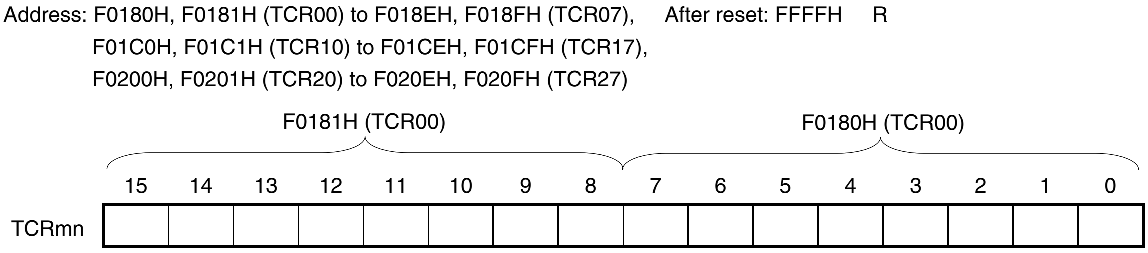
(1) Timer counter register mn (TCRmn)

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 6-6. Format of Timer Counter Register mn (TCRmn)



Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07), After reset: FFFFH R
F01C0H, F01C1H (TCR10) to F01CEH, F01CFH (TCR17),
F0200H, F0201H (TCR20) to F020EH, F020FH (TCR27)

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 (in case of TAU0), TAU1EN bit (in case of TAU1), or TAU2EN bit (in case of TAU2) of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

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

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

Caution The count value is not captured to TDRmn even when TCRmn is read.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

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 (TTm_n = 1)	Operation restart after count operation paused (TTm_n = 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 TSm_n has been set to “1” while TEm_n = 0 are shown. The read value is held in the TCRmn register until the count operation starts.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(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 6-7. Format of Timer Data Register mn (TDRmn)

Address: FFF18H, FFF19H (TDR00), FFF1AH, FFF1BH (TDR01), After reset: 0000H R/W
FFF64H, FFF65H (TDR02) to FFF6EH, FFF6FH (TDR07),
FFF70H, FFF71H (TDR10) to FFF7EH, FFF7FH (TDR17),
FFF90H, FFF91H (TDR20) to FFF9EH, FFF9FH (TDR27)

FFF19H (TDR00) FFF18H (TDR00)

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

(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 TDRmn is used as capture register

The count value of TCRmn is captured to TDRmn when the capture trigger is input.

A valid edge of the TImn pin can be selected as the capture trigger. This selection is made by TMRmn.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer output enable register m (TOEm)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Noise filter enable registers 1 to 3 (NFEN1 to NFEN3)
- Timer input select registers 0, 1 (TIS0, TIS1)
- Timer output select registers 0, 1 (TOS0, TOS1)
- Serial communication pin select register (STSEL)
- Port mode registers 0, 1, 3 to 7, 12, 15 (PM0, PM1, PM3 to PM7, PM12, PM15)
- Port registers 0, 1, 3 to 7, 12, 15 (P0, P1, P3 to P7, P12, P15)

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3, mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(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.

When the timer array unit 2 is used, be sure to set bit 2 (TAU2EN) 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 6-8. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	ADCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN

TAUmEN	Control of timer array unit m 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.

Cautions 1. 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 (except for timer input select register m (TISm), timer output select register m (TOSm), noise filter enable registers 1 to 3 (NFEN1 to NFEN3), serial communication pin select register (STSEL), port mode registers 0, 1, 3 to 7, 12, 15 (PM0, PM1, PM3 to PM7, PM12, PM15), and port registers 0, 1, 3 to 7, 12, 15 (P0, P1, P3 to P7, P12, P15)).

2. In the 78K0R/HC3, be sure to clear bit 2 to "0".

Remark m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3

(2) Timer clock select register m (TPSm)

TPSm is a 16-bit register that is used to select four types of operation clocks (CKm0 to CKm3) that are commonly supplied to each channel. CKm3 is selected by bits 15 to 12 of TPSm, CKm2 is selected by bits 11 to 8 of TPSm, 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 CKSmn0 = 0 and CKSmn1 = 0 are in the operation stopped state (TEm_n = 0)
Rewriting of PRSm10 to PRSm13 bits:	Possible only when all the channels set to CKSmn0 = 1 and CKSmn1 = 0 are in the operation stopped state (TEm_n = 0)
Rewriting of PRSm20 to PRSm23 bits:	Possible only when all the channels set to CKSmn0 = 0 and CKSmn1 = 1 are in the operation stopped state (TEm_n = 0)
Rewriting of PRSm30 to PRSm33 bits:	Possible only when all the channels set to CKSmn0 = 1 and CKSmn1 = 1 are in the operation stopped state (TEm_n = 0)

TPSm can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-9. Format of Timer Clock Select Register m (TPSm)

Address: F01B6H, F01B7H (TPS0), F01F6H, F01F7H (TPS1) After reset: 0000H R/W
F0236H, F0237H (TPS2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	PRS m33	PRS m32	PRS m31	PRS m30	PRS m23	PRS m22	PRS m21	PRS m20	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operation clock (CKmk)[Note]				
					f_{CLK} = 2 MHz	f_{CLK} = 6 MHz	f_{CLK} = 12 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	2 MHz	6 MHz	12 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	3 MHz	6 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.5 MHz	3 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	750 kHz	1.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	375 kHz	750 kHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	187.5 kHz	375 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.25 kHz	93.75 kHz	187.5 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.62 kHz	46.87 kHz	93.75 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	23.43 kHz	46.87 kHz	93.75 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	11.71 kHz	23.43 kHz	46.87 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	5.85 kHz	11.71 kHz	23.43 kHz
1	0	1	1	$f_{CLK}/2^{11}$	976 Hz	2.92 kHz	5.85 kHz	11.71 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.46 kHz	2.92 kHz	5.85 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	732.42 Hz	1.46 kHz	2.92 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	366.21 Hz	732.42 Hz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	183.10 Hz	366.21 Hz	732.42 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop the timer array unit (TTm = 00FFH).
The timer array unit must also be stopped if the operating clock specified by using the CKSmn bit (f_{MCK}), or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
k = 0 to 3
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(3) Timer mode register mn (TMRmn)

TMRmn sets an operation mode of channel n. It is used to select an operation clock (f_{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 6-10. Format of Timer Mode Register mn (TMRmn) (1/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07), After reset: 0000H R/W
F01D0H, F01D1H (TMR10) to F01DEH, F01DFH (TMR17),
F0210H, F0211H (TMR20) to F021EH, F021FH (TMR27)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKS mn1	CKS mn0	0	CCS mn0	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CKS mn1	CKS mn0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CKm0 set byTPSm register
0	1	Operation clock CKm1 set by TPSm register
1	0	Operation clock CKm2 set by TPSm register
1	1	Operation clock CKm3 set by TPSm register
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) is generated depending on the setting of the CCSmn bit.		

CCS mn0	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by CKSmn bit
1	Valid edge of input signal input from TImn pin
Count clock (f_{TCLK}) is used for the timer counter, output controller, and interrupt controller.	

Cautions 1. Be sure to clear bits 13, 5, and 4 to "0".

2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for f_{CLK} is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKSmn bit (f_{MCK}), or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-10. Format of Timer Mode Register mn (TMRmn) (2/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07), After reset: 0000H R/W
F01D0H, F01D1H (TMR10) to F01DEH, F01DFH (TMR17),
F0210H, F0211H (TMR20) to F021EH, F021FH (TMR27)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKS mn1	CKS mn0	0	CCS mn0	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

MAS TER mn	Selection of operation in single-operation function or as slave channel in combination-operation function /operation as master channel in combination-operation function of channel n
0	Operates in single-operation function or as slave channel in combination-operation function.
1	Operates as master channel in 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 single-operation function.	

STS mn2	STS mn1	STS mn0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of TImn pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of TImn pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination operation function).
Other than above			Setting prohibited

CIS mn1	CIS mn0	Selection of TImn pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STSmn2 to STSmn0 bits is other than 010B, set the CISmn1 to CISmn0 bits to 10B.		

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-10. Format of Timer Mode Register mn (TMRmn) (3/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07), After reset: 0000H R/W
F01D0H, F01D1H (TMR10) to F01DEH, F01DFH (TMR17),
F0210H, F0211H (TMR20) to F021EH, F021FH (TMR27)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKS mn1	CKS mn0	0	CCS mn0	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 table below).						

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) • Capture mode (0, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
• 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 (TSm_n = 1) is issued during operation, the counter is cleared, an interrupt is generated, and recounting is started.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(4) Timer status register mn (TSRmn)

TSRmn indicates the overflow status of the counter of channel n.

TSRmn is valid only in the capture mode (MDmn3 to MDmn1 = 010B) and capture & one-count mode (MDmn3 to MDmn1 = 110B). It will not be set in any other mode. See Table 6-3 for the operation of the OVF bit in each operation mode and set/clear conditions.

TSRmn can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TSRmn can be set with an 8-bit memory manipulation instruction with TSRmnL.

Reset signal generation clears this register to 0000H.

Figure 6-11. Format of Timer Status Register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07), After reset: 0000H R
F01E0H, F01E1H (TSR10) to F01EEH, F01EFH (TSR17),
F0220H, F0221H (TSR20) to F022EH, F022FH (TSR27),

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVF

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

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Table 6-3. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer Operation Mode	OVF	Set/Clear Conditions
• Capture mode • Capture & one-count mode	clear	When no overflow has occurred upon capturing
	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	– (Use prohibited, not set/cleared)
	set	

Remark The OVF 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 read with a 1-bit or 8-bit memory manipulation instruction with TEmL.

Reset signal generation clears this register to 0000H.

<R> **Figure 6-12. Format of Timer Channel Enable Status Register m (TEm)**

Address: F01B0H, F01B1H (TE0), F01E0H, F01F1H (TE1), After reset: 0000H R
F0230H, F0231H (TE2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	0	0	0	0	TEm_7	TEm_6	TEm_5	TEm_4	TEm_3	TEm_2	TEm_1	TEm_0

TE m_n	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(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 (TSm_n) of this register is set to 1, the corresponding bit (TEm_n) of timer channel enable status register m (TEm) is set to 1. TSmn is a trigger bit and cleared immediately when TEm_n = 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.

<R> **Figure 6-13. Format of Timer Channel Start Register m (TSm)**

Address: F01B2H, F01B3H (TS0), F01F2H, F01F3H (TS1), After reset: 0000H R/W
F0232H, F0233H (TS2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	0	0	0	0	TSm_7	TSm_6	TSm_5	TSm_4	TSm_3	TSm_2	TSm_1	TSm_0

TS m_n	Operation enable (start) trigger of channel n
0	No trigger operation
1	TEm_n 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 With the 78K0R/HC3, be sure to clear bits 15 to 8 of TS0 and TS1 to 0.
With the 78K0R/HE3 and 78K0R/HF3, be sure to clear bits 15 to 8 of TS0 and TS1, and bits 15 to 4 of TS2 to 0.
With the 78K0R/HG3, be sure to clear bits 15 to 8 of TS0 to TS2 to 0.

Remarks 1. When the TSm register is read, 0 is always read.

2. m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Table 6-4. Operations from Count Operation Enabled State to TCRmn Count Start

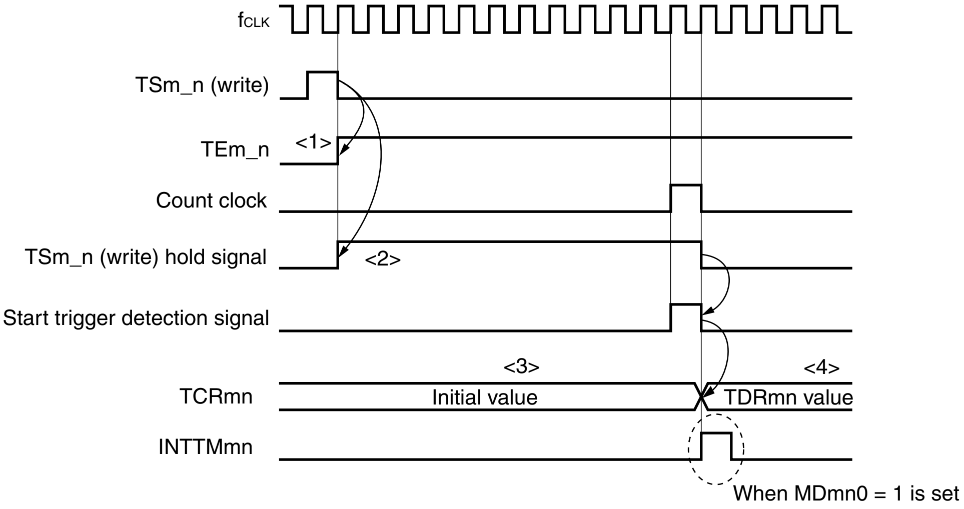
Timer Operation Mode	Operation When TSm_n = 1 Is Set
• Interval timer mode	No operation is carried out from start trigger detection (TSm_n=1) until count clock generation. The first count clock loads the value of TDRmn to TCRmn and the subsequent count clock performs count down operation (see **6.3 (6) (a) Start timing in interval timer mode**).
• Event counter mode	Writing 1 to TSm_n 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 TSm_n = 0, writing 1 to TSm_n 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 TSm_n = 0, writing 1 to TSm_n 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**).

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(a) Start timing in interval timer mode

<1> Writing 1 to TSm_n sets TEm_n = 1.
<2> The write data to TSm_n 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.

Figure 6-14. Start Timing (In Interval Timer Mode)



Caution In the first cycle operation of count clock after writing TSm_n, 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 m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(b) Start timing in event counter mode

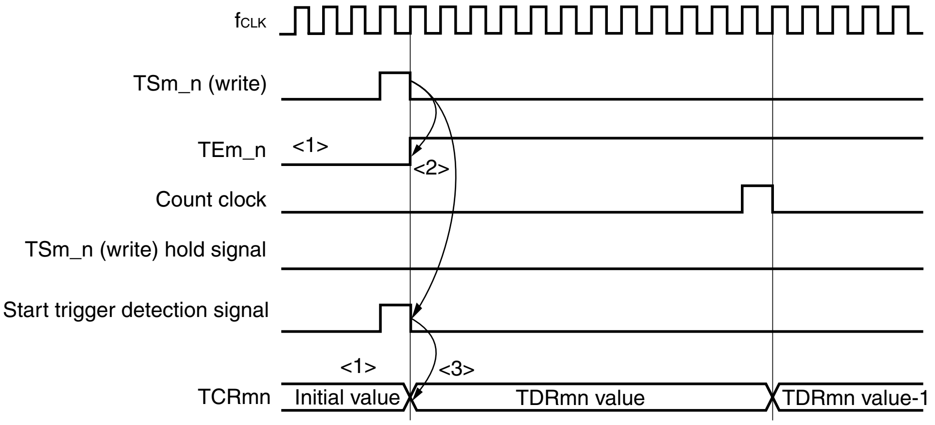
<1> While TEm_n is set to 0, TCRmn holds the initial value.

<2> Writing 1 to TSm_n sets 1 to TEm_n.

<3> As soon as 1 has been written to TSm_n and 1 has been set to TEm_n, the “TDRmn value” is loaded to TCRmn to start counting.

<4> After that, the TCRmn value is counted down according to the count clock.

Figure 6-15. Start Timing (In Event Counter Mode)

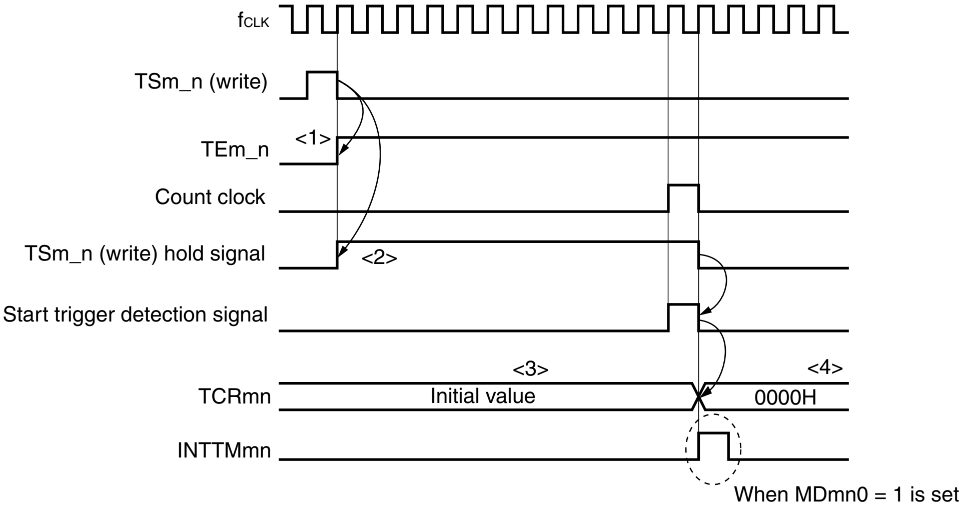


Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(c) Start timing in capture mode

<1> Writing 1 to TSm_n sets TEm_n = 1.
<2> The write data to TSm_n is held until count clock generation.
<3> TCRmn holds the initial value until count clock generation.
<4> On generation of count clock, 0000H is loaded to TCRmn and count starts.

Figure 6-16. Start Timing (In Capture Mode)



Caution In the first cycle operation of count clock after writing TSm_n, 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.

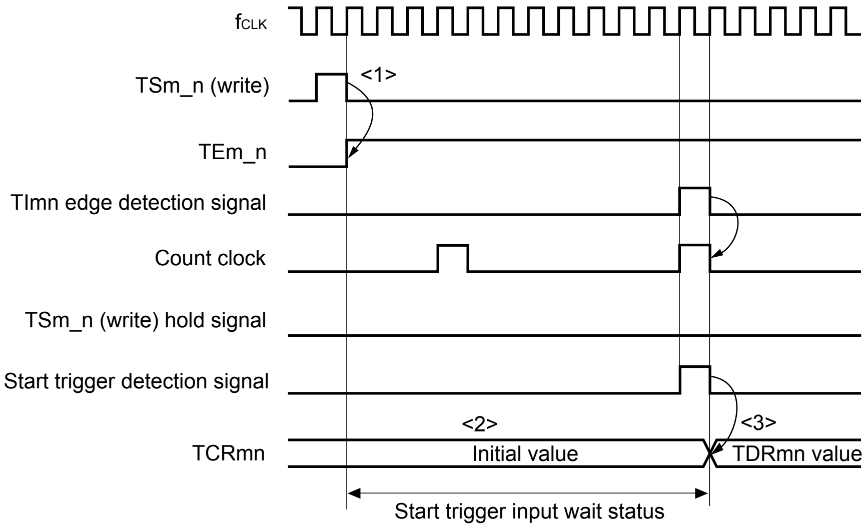
Remark m: Unit number (m = 0, 1:, 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(d) Start timing in one-count mode

<1> Writing 1 to TSm_n sets TEm_n = 1.
<2> Enters the start trigger input wait status, and TCRmn holds the initial value.
<3> On start trigger detection, the "TDRmn value" is loaded to TCRmn and count starts.

<R>

Figure 6-17. Start Timing (In One-count Mode)


fCLK
TSm_n (write)
<1>
TEm_n
TImn edge detection signal
Count clock
TSm_n (write) hold signal
Start trigger detection signal
<2>
<3>
TCRmn
Initial value
TDRmn value
Start trigger input wait status

Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TImn is used).

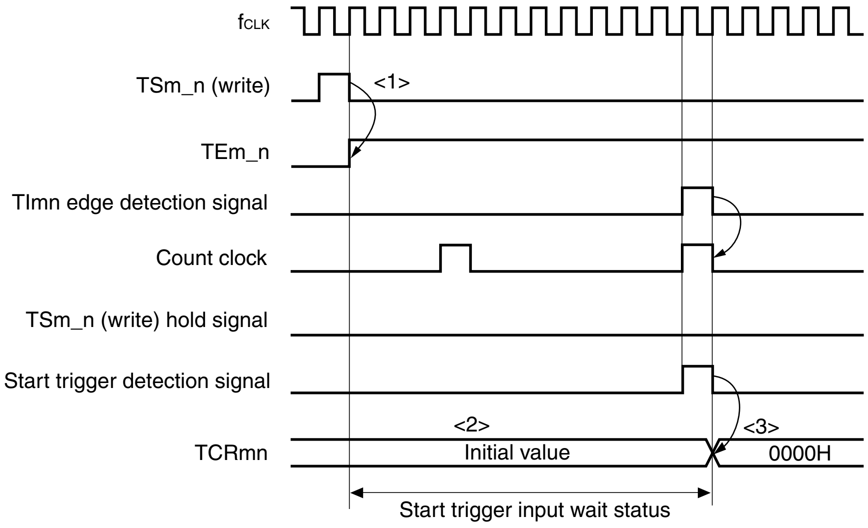
Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(e) Start timing in capture & one-count mode

<1> Writing 1 to TSm_n sets TEm_n = 1.
<2> Enters the start trigger input wait status, and TCRmn holds the initial value.
<3> On start trigger detection, 0000H is loaded to TCRmn and count starts.

<R>

Figure 6-18. Start Timing (In Capture & One-count Mode)


f_{CLK}
TSm_n (write)
<1>
TEm_n
TImn edge detection signal
Count clock
TSm_n (write) hold signal
Start trigger detection signal
<2>
<3>
TCRmn
Initial value
0000H
Start trigger input wait status

Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TImn is used).

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(7) Timer channel stop register m (TTm)

TTm is a trigger register that is used to stop the counting operation of each channel.

When a bit (TTm_n) of this register is set to 1, the corresponding bit (TEm_n) of timer channel enable status register m (TEm) is cleared to 0. TTm_n is a trigger bit and cleared to 0 immediately when TEm_n = 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.

<R> **Figure 6-19. Format of Timer Channel Stop Register m (TTm)**

Address: F01B4H, F01B5H (TT0), F01F4H, F01F5H (TT1) After reset: 0000H R/W
F0234H, F0235H (TT2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTm	0	0	0	0	0	0	0	0	TTm_7	TTm_6	TTm_5	TTm_4	TTm_3	TTm_2	TTm_1	TTm_0

TT m_n	Operation stop trigger of channel n
0	No trigger operation
1	Operation is stopped (stop trigger is generated).

Caution With the 78K0R/HC3, be sure to clear bits 15 to 8 of TT0 and TT1 to 0.
With the 78K0R/HE3 and 78K0R/HF3, be sure to clear bits 15 to 8 of TT0 and TT1, and bits 15 to 4 of TT2 to 0.
With the 78K0R/HG3, be sure to clear bits 15 to 8 of TT0 to TT2 to 0.

Remarks 1. When the TTm register is read, 0 is always read.

2. m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(8) Timer output enable register m (TOEm)

TOEm is used to enable or disable timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TOmn bit of the timer output register (TOm) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TOmn).

TOEm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOEm can be set with a 1-bit or 8-bit memory manipulation instruction with TOEmL.

Reset signal generation clears this register to 0000H.

<R> **Figure 6-20. Format of Timer Output Enable Register m (TOEm)**

Address: F01BAH, F01BBH (TOE0), F01FAH, F01FBH (TOE1), After reset: 0000H R/W
F023AH, F023BH (TOE2),

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEm	0	0	0	0	0	0	0	0	TOE m_7	TOE m_6	TOE m_5	TOE m_4	TOE m_3	TOE m_2	TOE m_1	TOE m_0

TOE m_n	Timer output enable/disable of channel n
0	The TOmn operation stopped by count operation (timer channel output bit). Writing to the TOmn bit is enabled. The TOmn pin functions as data output, and it outputs the level set to the TOmn bit. The output level of the TOmn pin can be manipulated by software.
1	The TOmn operation enabled by count operation (timer channel output bit). Writing to the TOmn bit is disabled (writing is ignored). The TOmn pin functions as timer output, and the TOEm_n is set or reset depending on the timer operation. The TOmn pin outputs the square-wave or PWM depending on the timer operation.

Caution With the 78K0R/HC3, be sure to clear bits 15 to 8 of TOE0 and TOE1 to 0.
With the 78K0R/HE3 and 78K0R/HF3, be sure to clear bits 15 to 8 of TOE0 and TOE1, and bits 15 to 4 of TOE2 to 0.
With the 78K0R/HG3, be sure to clear bits 15 to 8 of TOE0 to TOE2 to 0.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(9) Timer output register m (TOm)

TOm is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TOmn) of each channel.

This register can be rewritten by software only when timer output is disabled (TOEm_n = 0). When timer output is enabled (TOEm_n = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the pins sharing timer output as port function pins, set the corresponding TOmn bit to "0".

See **2.1 (2) Non-port pins** for details of the pins sharing timer output.

TOm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOm can be set with an 8-bit memory manipulation instruction with TOmL.

Reset signal generation clears this register to 0000H.

Figure 6-21. Format of Timer Output Register m (TOm)

Address: F01B8H, F01B9H (TO0), F01F8H, F01F9H (TO1), After reset: 0000H R/W
F0238H, F0239H (TO2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOm	0	0	0	0	0	0	0	0	TOm 7	TOm 6	TOm 5	TOm 4	TOm 3	TOm 2	TOm 1	TOm 0

TO mn	Timer output of channel n
0	Timer output value is "0".
1	Timer output value is "1".

Caution With the 78K0R/HC3, be sure to clear bits 15 to 8 of TO0 and TO1 to 0.
With the 78K0R/HE3 and 78K0R/HF3, be sure to clear bits 15 to 8 of TO0 and TO1, and bits 15 to 4 of TO2 to 0.
With the 78K0R/HG3, be sure to clear bits 15 to 8 of TO0 to TO2 to 0.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(10) Timer output level register m (TOLm)

TOLm is a register that controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOEm_n = 1) in the combination operation mode (TOMmn = 1). In the toggle mode (TOMmn = 0), this register setting is invalid.

TOLm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOLm can be set with an 8-bit memory manipulation instruction with TOLmL.

Reset signal generation clears this register to 0000H.

Figure 6-22. Format of Timer Output Level Register m (TOLm)

Address: F01BCH, F01BDH (TOL0), F01FCH, F01FDH (TOL1), After reset: 0000H R/W
F023CH, F023DH (TOL2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOLm	0	0	0	0	0	0	0	0	TOL m7	TOL m6	TOL m5	TOL m4	TOL m3	TOL m2	TOL m1	TOL m0

TOL mn	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Inverted output (active-low)

Caution With the 78K0R/HC3, be sure to clear bits 15 to 8 of TOL0 and TOL1 to 0.
With the 78K0R/HE3 and 78K0R/HF3, be sure to clear bits 15 to 8 of TOL0 and TOL1, and bits 15 to 4 of TOL2 to 0.
With the 78K0R/HG3, be sure to clear bits 15 to 8 of TOL0 to TOL2 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. m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(11) Timer output mode register m (TOMm)

TOMm is used to control the timer output mode of each channel.

When a channel is used for the single-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, one-shot pulse output, or multiple PWM output), set the corresponding bit of the master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOEm_n = 1).

TOMm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOMm can be set with an 8-bit memory manipulation instruction with TOMmL.

Reset signal generation clears this register to 0000H.

Figure 6-23. Format of Timer Output Mode Register m (TOMm)

Address: F01BEH, F01BFH (TOM0), F01FEH, F01FFH (TOM1), After reset: 0000H R/W
F023EH, F023FH (TOM2),

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOMm	0	0	0	0	0	0	0	0	TOM m7	TOM m6	TOM m5	TOM m4	TOM m3	TOM m2	TOM m1	TOM m0

TOM mn	Control of timer output mode of channel n
0	Toggle mode (to produce toggle output by timer interrupt request signal (INTTMmn))
1	Combination operation mode (output is set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTMmp) of the slave channel.)

Caution With the 78K0R/HC3, be sure to clear bits 15 to 8 of TOM0 and TOM1 to 0.
With the 78K0R/HE3 and 78K0R/HF3, be sure to clear bits 15 to 8 of TOM0 and TOM1, and bits 15 to 4 of TOM2 to 0.
With the 78K0R/HG3, be sure to clear bits 15 to 8 of TOM0 to TOM2 to 0.

Remark m: Unit number, n: Channel number, p: Slave channel number
When m = 0
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
n < p ≤ 7 (where p is a consecutive integer greater than n)
When m = 1
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
n < p ≤ 7 (where p is a consecutive integer greater than n)
When m = 2
n = 0 to 3 (n = 0, 2 for master channel): 78K0R/HE3, 78K0R/HF3
n < p ≤ 3 (where p is a consecutive integer greater than n)
n = 0 to 7 (n = 0, 2, 4, 6 for master channel): 78K0R/HG3
n < p ≤ 7 (where p is a consecutive integer greater than n)

(12) Noise filter enable registers 1 to 3 (NFEN1 to NFEN3)

NFEN1 is used to set for each channel whether the noise filter can be used for the input signal from the timer input pin of timer array unit 0.

NFEN2 is used to set for each channel whether the noise filter can be used for the input signal from the timer input pin of timer array unit 1.

NFEN3 is used to set for each channel whether the noise filter can be used for the input signal from the timer input pin of timer array unit 2.

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, it detects the correspondence between the 2 clocks with the operation clock (f_{MCK}), and synchronizes them.

When the noise filter is OFF, only synchronization is performed with the operation clock (f_{MCK}).

NFEN1 to NFEN3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 6-24. 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	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN07	Enable/disable using noise filter of TI07 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN06	Enable/disable using noise filter of TI06 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN05	Enable/disable using noise filter of TI05 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN04	Enable/disable using noise filter of TI04 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00 pin input signal
0	Noise filter OFF
1	Noise filter ON

Remark The pins mounted differ depending on the product. See **6.3 (13) Timer input select registers 0, 1 (TIS0, TIS1)** for details.

Figure 6-25. Format of Noise Filter Enable Register 2 (NFEN2)

Address: F0062H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN2	TNFEN17	TNFEN16	TNFEN15	TNFEN14	TNFEN13	TNFEN12	TNFEN11	TNFEN10

TNFEN17	Enable/disable using noise filter of TI17 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN16	Enable/disable using noise filter of TI16 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN15	Enable/disable using noise filter of TI15 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN14	Enable/disable using noise filter of TI14 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN13	Enable/disable using noise filter of TI13 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN12	Enable/disable using noise filter of TI12 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN11	Enable/disable using noise filter of TI11 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN10	Enable/disable using noise filter of TI10 pin input signal
0	Noise filter OFF
1	Noise filter ON

Remark The pins mounted differ depending on the product. See **6.3 (13) Timer input select registers 0, 1 (TIS0, TIS1)** for details.

Figure 6-26. Format of Noise Filter Enable Register 3 (NFEN3) (78K0R/HE3, 78K0R/HF3, 78K0R/HG3)

Address: F0063H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN3	TNFEN27[Note]	TNFEN26[Note]	TNFEN25[Note]	TNFEN24[Note]	TNFEN23	TNFEN22	TNFEN21	TNFEN20

TNFEN27	Enable/disable using noise filter of TI27 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN26	Enable/disable using noise filter of TI26 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN25	Enable/disable using noise filter of TI25 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN24	Enable/disable using noise filter of TI24 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN23	Enable/disable using noise filter of TI23 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN22	Enable/disable using noise filter of TI22 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN21	Enable/disable using noise filter of TI21 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN20	Enable/disable using noise filter of TI20 pin input signal
0	Noise filter OFF
1	Noise filter ON

Note 78K0R/HG3 only. With the 78K0R/HE3 and 78K0R/HF3, be sure to clear these bits to 0.

<R> **(13) Timer input select registers 0, 1 (TIS0, TIS1)**

TIS0 and TIS1 select the input pins of timer array units 0 and 1 from two ports.

TIS0 and TIS1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Caution With TIS1_2 and TIS1_3, the signal to be input to the timer array unit is finally determined by a combination with the LIN function.
With TIS1_0, the signal to be input to the timer array unit is finally determined by a combination with the TM30K.

Figure 6-27. Format of Timer Input Select Register 0 (TIS0) (1/2)

Address: FFF3EH After reset: 00H R/W

(1) 78K0R/HC3, 78K0R/HE3

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	TIS0_5	0	0	0	0	0

(2) 78K0R/HF3

Symbol	7	6	5	4	3	2	1	0
TIS0	TIS0_7	TIS0_6	TIS0_5	TIS0_4	0	TIS0_2	TIS0_1	TIS0_0

(3) 78K0R/HG3

Symbol	7	6	5	4	3	2	1	0
TIS0	TIS0_7	TIS0_6	TIS0_5	TIS0_4	TIS0_3	TIS0_2	TIS0_1	TIS0_0

TIS0_7	TI07 input pin switch control
0	P41/TOOL1/TI07/TO07
1	P44/TI07/TO07

TIS0_6	TI06 input pin switch control
0	P14/LRxD0/INTPLR0/TI06/TO06
1	P02/TI06/TO06

TIS0_5	TI05 input pin switch control
0	P40/TOOL0/TI05/TO05
1	P00/INTP7/TI05/TO05

TIS0_4	TI04 input pin switch control
0	P13/LTxD0/TI04/TO04
1	P01/TI04/TO04

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Figure 6-27. Format of Timer Input Select Register 0 (TIS0) (2/2)

TIS0_3	TI03 input pin switch control
0	P125/INTP1/ADTRG/TI03/TO03
1	P127/TI03/TO03

TIS0_2	TI02 input pin switch control
0	P11/SI10/CRxD/LRxD1/INTPLR1/TI02/TO02
1	P67/TI02/TO02

TIS0_1	TI01 input pin switch control
0	P30/SSI00/INTP2/TI01/TO01
1	P126/TI01/TO01

TIS0_0	TI00 input pin switch control
0	P10/$\overline{\text{SCK10}}$/CTxD/LTxD1/TI00/TO00
1	P66/TI00/TO00

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Figure 6-28. Format of Timer Input Select Register 1 (TIS1) (1/2)

Address: FFF3FH After reset: 00H R/W

(1) 78K0R/HC3, 78K0R/HE3

Symbol	7	6	5	4	3	2	1	0
TIS1	0	0	0	0	0	0	TIS1_1	0

(2) 78K0R/HF3, 78K0R/HG3

Symbol	7	6	5	4	3	2	1	0
TIS1	TIS1_7	TIS1_6	TIS1_5	TIS1_4	TIS1_3	TIS1_2	TIS1_1	TIS1_0

TIS1_7	TI17 input pin switch control
0	P71/KR1/INTP6/TI17/TO17
1	P57/TI17/TO17

TIS1_6	TI16 input pin switch control
0	P12/SO10/INTP3/TI16/TO16
1	P65/TI16/TO16

TIS1_5	TI15 input pin switch control
0	P70/KR0/INTP5/TI15/TO15/LVIOUT
1	P56/TI15/TO15

TIS1_4	TI14 input pin switch control
0	P17/$\overline{\text{SCK00}}$/TI14/TO14
1	P64/TI14/TO14

TIS1_3	TI13 input pin switch control
0	P32/INTP4/TI13/TO13$^{\text{Note 1}}$
1	P55/TI13/TO13$^{\text{Note 1}}$

TIS1_2	TI12 input pin switch control
0	P16/SI00/TI12/TO12$^{\text{Note 2}}$
1	P46/TI12/TO12$^{\text{Note 2}}$

Notes 1. When STSEL.TMLIN1 is 1, this is the pin selected by using the STSLIN1 bit of the STSEL register.

2. When STSEL.TMLIN0 is 1, this is P14/LRxD0/INTPLR0/TI06/TO06.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Figure 6-28. Format of Timer Input Select Register 1 (TIS1) (2/2)

TIS1_1	TI11 input pin switch control	Remark
0	P54/TI11/TO11	78K0R/HF3, 78K0R/HG3
	P120/INTP0/EXLVI/TI11/TO11	78K0R/HC3, 78K0R/HE3
1	P31/INTP2/STOPST/TI11/TO11	–

TIS1_0	TI10 input pin switch control
0	P15/SO00/TI10/TO10[Note]
1	P45/TI10/TO10[Note]

Note The internal low-speed oscillation clock is selected for the TI10 input pin when STSEL.TM30K = 1.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

<R>

(14) Timer output select registers 0, 1 (TOS0, TOS1)

TOS0 and TOS1 select the output pins of timer array units 0 and 1 from two ports.

TOS0 and TOS1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 6-29. Format of Timer Output Select Register 0 (TOS0) (1/2)

Address: FFF60H After reset: 00H R/W

(1) 78K0R/HC3, 78K0R/HE3

Symbol	7	6	5	4	3	2	1	0
TOS0	0	0	TOS0_5	0	0	0	0	0

(2) 78K0R/HF3

Symbol	7	6	5	4	3	2	1	0
TOS0	TOS0_7	TOS0_6	TOS0_5	TOS0_4	0	TOS0_2	TOS0_1	TOS0_0

(3) 78K0R/HG3

Symbol	7	6	5	4	3	2	1	0
TOS0	TOS0_7	TOS0_6	TOS0_5	TOS0_4	TOS0_3	TOS0_2	TOS0_1	TOS0_0

TOS0_7	TO07 output pin switch control
0	P41/TOOL1/TI07/TO07
1	P44/TI07/TO07

TOS0_6	TO06 output pin switch control
0	P14/LRxD0/INTPLR0/TI06/TO06
1	P02/TI06/TO06

TOS0_5	TO05 output pin switch control
0	P40/TOOL0/TI05/TO05
1	P00/TI05/TO05/INTP7

TOS0_4	TO04 output pin switch control
0	P13/LTxD0/TI04/TO04
1	P01/TI04/TO04

TOS0_3	TO03 output pin switch control
0	P125/INTP1/ADTRG/TI03/TO03
1	P127/TI03/TO03

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Figure 6-29. Format of Timer Output Select Register 0 (TOS0) (2/2)

TOS0_2	TO02 output pin switch control
0	P11/SI10/CRxD/LRxD1/INTPLR1/TI02/TO02
1	P67/TI02/TO02

TOS0_1	TO01 output pin switch control
0	P30/$\overline{\text{SSI00}}$/INTP2/TI01/TO01
1	P126/TI01/TO01

TOS0_0	TO00 output pin switch control
0	P10/$\overline{\text{SCK10}}$/CTxD/LTxD1/TI00/TO00
1	P66/TI00/TO00

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Figure 6-30. Format of Timer Output Select Register 1 (TOS1)

Address: F0061H After reset: 00H R/W

(1) 78K0R/HC3, 78K0R/HE3

Symbol	7	6	5	4	3	2	1	0
TOS1	0	0	0	0	0	0	TOS1_1	0

(2) 78K0R/HF3, 78K0R/HG3

Symbol	7	6	5	4	3	2	1	0
TOS1	TOS1_7	TOS1_6	TOS1_5	TOS1_4	TOS1_3	TOS1_2	TOS1_1	TOS1_0

TOS1_7	TO17 output pin switch control
0	P71/KR1/INTP6/TI17/TO17
1	P57/TI17/TO17

TOS1_6	TO16 output pin switch control
0	P12/SO10/INTP3/TI16/TO16
1	P65/TI16/TO16

TOS1_5	TO15 output pin switch control
0	P70/KR0/INTP5/TI15/TO15/LVIOUT
1	P56/TI15/TO15

TOS1_4	TO14 output pin switch control
0	P17/SCK00/TI14/TO14
1	P64/TI14/TO14

TOS1_3	TO13 output pin switch control
0	P32/INTP4/TI13/TO13
1	P55/TI13/TO13

TOS1_2	TO12 output pin switch control
0	P16/SI00/TI12/TO12
1	P46/TI12/TO12

TOS1_1	TO11 output pin switch control	Remark
0	P54/TI11/TO11	78K0R/HF3, 78K0R/HG3
	P120/INTP0/EXLVI/TI11/TO11	78K0R/HC3, 78K0R/HE3
1	P31/INTP2/STOPST/TI11/TO11	–

TOS1_0	TO10 output pin switch control
0	P15/SO00/TI10/TO10
1	P45/TI10/TO10

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

(15) Serial communication pin select register (STSEL)

The STSEL register is used to switch between the input source of the timer array unit and the communication pins of LIN-UARTn and the serial array unit.

This register can be read or written in 1-bit units or 8-bit units.

Figure 6-31. Format of Serial Communication Pin Select Register (STSEL)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
STSEL	STSLIN1	TMCAN	STSIIC11	STSCSI00	TM30K	TMLIN1	TMLIN0	0

TMCAN	Channel 4 input source switch control of timer array unit 1
0	TI14 pin input (pin input selected by TI14 bit)
1	TSOUTPUT input (CAN time stamp function)

TM30K	Channel 0 input source switch control of timer array unit 1
0	TI10 pin input (pin input selected by TIS1_0 bit)
1	Internal low-speed oscillator output

TMLIN1	Channel 3 input source switch control of timer array unit 1
0	TI13 pin input (pin input selected by TIS1_3 bit)
1	LRxD1 pin input

TMLIN0	Channel 2 input source switch control of timer array unit 1
0	TI12 pin input (pin input selected by TIS1_2 bit)
1	LRxD0 pin input

Remark During LIN communication with LIN-UART, when in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B), select the input signal of the serial data input pin (LRxDn) as the timer input by setting TMLINn to 1.

(16) Port mode registers 0, 1, 3 to 7, 12, 15 (PM0, PM1, PM3 to PM7, PM12, PM15)

These registers set input/output of ports 0, 1, 3 to 7, 12, and15 in 1-bit units.

When using the pins as timer outputs or timer inputs, set the port register and port mode register as shown in Table 6-5.

PM0, PM1, PM3 to PM7, PM12, and PM15 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Table 6-5. Port Register and Port Mode Register Settings When Using Timer (1/3)

Pin Name	Pin Setting	Port Register	Port Mode Register
P00/TI05/TO05/INTP7	Timer output	0	0
	Timer input	×	1
P01/TI04/TO04	Timer output	0	0
	Timer input	×	1
P02/TI06/TO06	Timer output	0	0
	Timer input	×	1
P10/$\overline{\text{SCK10}}$/CTxD/LTxD1/TI00/TO00	Timer output	0	0
	Timer input	×	1
P11/SI10/CRxD/LRxD1/INTPLR1/TI02/TO02	Timer output	0	0
	Timer input	×	1
P12/SO10/INTP3/TI16/TO16	Timer output	0	0
	Timer input	×	1
P13/LTxD0/TI04/TO04	Timer output	0	0
	Timer input	×	1
P14/LRxD0/INTPLR0/TI06/TO06	Timer output	0	0
	Timer input	×	1
P15/SO00/TI10/TO10	Timer output	0	0
	Timer input	×	1
P16/SI00/TI12/TO12	Timer output	0	0
	Timer input	×	1
P17/$\overline{\text{SCK00}}$/TI14/TO14	Timer output	0	0
	Timer input	×	1
P30/$\overline{\text{SSI00}}$/INTP2/TI01/TO01	Timer output	0	0
	Timer input	×	1
P31/INTP2/STOPST/TI11/TO11	Timer output	0	0
	Timer input	×	1
P32/INTP4/TI13/TO13	Timer output	0	0
	Timer input	×	1
P40/TOOL0/TI05/TO05	Timer output	0	0
	Timer input	×	1

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark X: Don’t care

Table 6-5. Port Register and Port Mode Register Settings When Using Timer (2/3)

Pin Name	Pin Setting	Port Register	Port Mode Register
P41/TOOL1/TI07/TO07	Timer output	0	0
	Timer input	×	1
P44/TI07/TO07	Timer output	0	0
	Timer input	×	1
P45/TI10/TO10	Timer output	0	0
	Timer input	×	1
P46/TI12/TO12	Timer output	0	0
	Timer input	×	1
P50/INTP3/TI20/TO20	Timer output	0	0
	Timer input	×	1
P51/TI21/TO21	Timer output	0	0
	Timer input	×	1
P52/STOPST/TI22/TO22	Timer output	0	0
	Timer input	×	1
P53/TI23/TO23	Timer output	0	0
	Timer input	×	1
P54/TI11/TO11	Timer output	0	0
	Timer input	×	1
P55/TI13/TO13	Timer output	0	0
	Timer input	×	1
P56/TI15/TO15	Timer output	0	0
	Timer input	×	1
P57/TI17/TO17	Timer output	0	0
	Timer input	×	1
P64/TI14/TO14	Timer output	0	0
	Timer input	×	1
P65/TI16/TO16	Timer output	0	0
	Timer input	×	1
P66/TI00/TO00	Timer output	0	0
	Timer input	×	1
P67/TI02/TO02	Timer output	0	0
	Timer input	×	1
P70/KR0/INTP5/TI15/TO15/LVIOUT	Timer output	0	0
	Timer input	×	1
P71/KR1/INTP6/TI17/TO17	Timer output	0	0
	Timer input	×	1
P120/INTP0/EXLVI/TI11/TO11[Note]	Timer output	0	0
	Timer input	×	1

Note TI11, TO11 pins are 78K0R/HC3, 78K0R/HE3 only.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark X: Don't care

Table 6-5. Port Register and Port Mode Register Settings When Using Timer (3/3)

Pin Name	Pin Setting	Port Register	Port Mode Register
P125/INTP1/ADTRG/TI03/TO03	Timer output	0	0
	Timer input	×	1
P126/TI01/TO01	Timer output	0	0
	Timer input	×	1
P127/TI03/TO03	Timer output	0	0
	Timer input	×	1
P154/TI24/TO24	Timer output	0	0
	Timer input	×	1
P155/TI25/TO25	Timer output	0	0
	Timer input	×	1
P156/TI26/TO26	Timer output	0	0
	Timer input	×	1
P157/TI27/TO27	Timer output	0	0
	Timer input	×	1

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark X: Don’t care

<R>

Figure 6-32. Format of Port Mode Registers 0, 1, 3 to 7, 12, 15 (PM0, PM1, PM3 to PM7, PM12, PM15)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	1	PM0_3	PM0_2	PM0_1	PM0_0

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM1_7	PM1_6	PM1_5	PM1_4	PM1_3	PM1_2	PM1_1	PM1_0

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	PM3_2	PM3_1	PM3_0

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	PM4_7	PM4_6	PM4_5	PM4_4	PM4_3	PM4_2	PM4_1	PM4_0

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM5_7	PM5_6	PM5_5	PM5_4	PM5_3	PM5_2	PM5_1	PM5_0

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	PM6_7	PM6_6	PM6_5	PM6_4	PM6_3	PM6_2	PM6_1	PM6_0

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM7_7	PM7_6	PM7_5	PM7_4	PM7_3	PM7_2	PM7_1	PM7_0

Address: FFF2CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM12	PM12_7	PM12_6	PM12_5	1	1	1	1	PM12_0

Address: FFF2FH After reset: FFH R/W

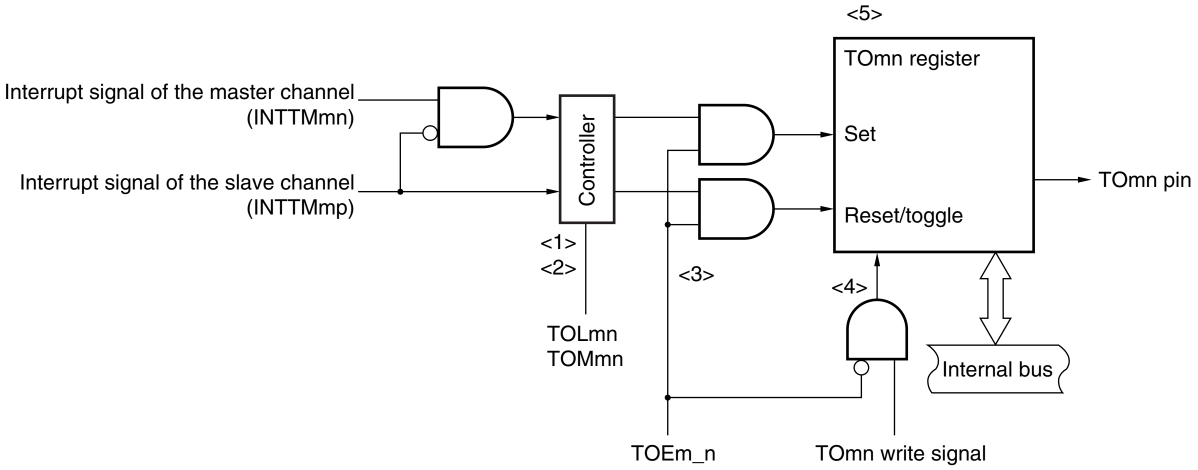
Symbol	7	6	5	4	3	2	1	0
PM15	PM15_7	PM15_6	PM15_5	PM15_4	PM15_3	PM15_2	PM15_1	PM15_0

PMm_n	Pmn pin I/O mode selection (m = 0, 1, 3 to 7, 12, 15; n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

6.4 Channel Output (TOmn Pin) Control

6.4.1 TOmn pin output circuit configuration

Figure 6-33. Output Circuit Configuration



The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (toggle mode), the set value of the TOLmn register is ignored and only INTTMmp (slave channel timer interrupt) is transmitted to the TOmn register.

<2> When TOMmn = 1 (combination operation mode), both INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are transmitted to the TOmn register.
At this time, the TOLmn register becomes valid and the signals are controlled as follows:

When TOLmn = 0: Forward operation (INTTMmn → set, INTTMmp → reset)
When TOLmn = 1: Reverse operation (INTTMmn → reset, INTTMmp → set)

When INTTMmn and INTTMmp are simultaneously generated, (0% output of PWM), INTTMmp (reset signal) takes priority, and INTTMmn (set signal) is masked.

Remark m: Unit number, n: Channel number, p: Slave channel number
When m = 0
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
n < p ≤ 7 (where p is a consecutive integer greater than n)
When m = 1
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
n < p ≤ 7 (where p is a consecutive integer greater than n)
When m = 2
n = 0 to 3 (n = 0, 2 for master channel): 78K0R/HE3, 78K0R/HF3
n < p ≤ 3 (where p is a consecutive integer greater than n)
n = 0 to 7 (n = 0, 2, 4, 6 for master channel): 78K0R/HG3
n < p ≤ 7 (where p is a consecutive integer greater than n)

<3> When TOEm_n = 1, INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are transmitted to the TOmn register. Writing to the TOmn register (TOmn write signal) becomes invalid.
When TOEm_n = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set TOEm_n = 0 and to write a value to TOmn.

<4> When TOEm_n = 0, writing to TOmn bit to the target channel (TOmn write signal) becomes valid. When TOEm_n = 0 neither INTTMmn (master channel timer interrupt) nor INTTMmp (slave channel timer interrupt) is transmitted to TOmn register.

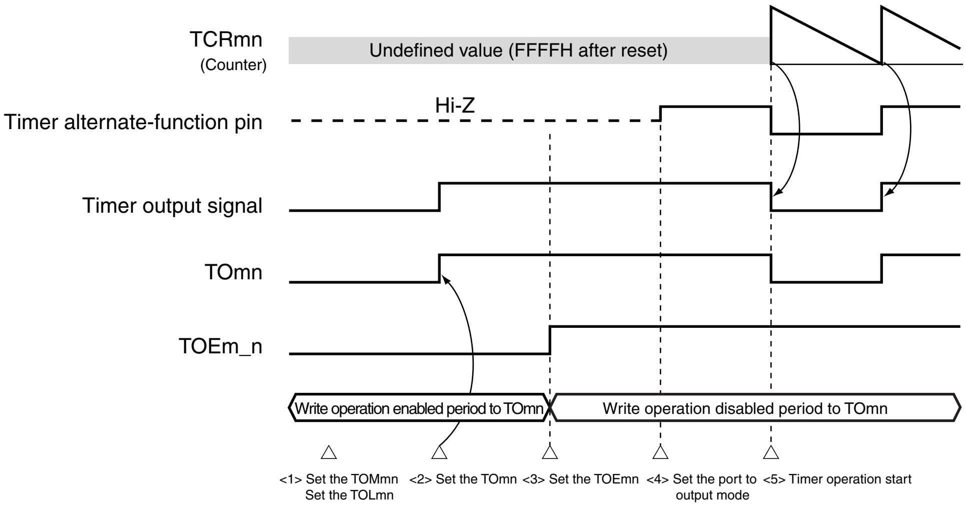
<5> The TOmn register can always be read, and the TOmn pin output level can be checked.

Remark m: Unit number, n: Channel number, p: Slave channel number
When m = 0
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
n < p ≤ 7 (where p is a consecutive integer greater than n)
When m = 1
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
n < p ≤ 7 (where p is a consecutive integer greater than n)
When m = 2
n = 0 to 3 (n = 0, 2 for master channel): 78K0R/HE3, 78K0R/HF3
n < p ≤ 3 (where p is a consecutive integer greater than n)
n = 0 to 7 (n = 0, 2, 4, 6 for master channel): 78K0R/HG3
n < p ≤ 7 (where p is a consecutive integer greater than n)

6.4.2 TOmn pin output setting

The following figure shows the procedure and status transition of TOmn output pin from initial setting to timer operation start.

Figure 6-34. Status Transition from Timer Output Setting to Operation Start


TCRmn
(Counter)
Undefined value (FFFFH after reset)
Timer alternate-function pin
Hi-Z
Timer output signal
TOmn
TOEm_n
Write operation enabled period to TOmn
Write operation disabled period to TOmn
<1> Set the TOMmn Set the TOLmn
<2> Set the TOmn
<3> Set the TOEmn
<4> Set the port to output mode
<5> Timer operation start

<1> The operation mode of timer output is set.

- TOMmn bit (0: Toggle mode, 1: Combination operation mode)
- TOLmn bit (0: Forward output, 1: Reverse output)

<2> The timer output signal is set to the initial status by setting TOmn.
<3> The timer output operation is enabled by writing 1 to TOEm_n (writing to TOmn is disabled).
<4> The port I/O setting is set to output (see **6.3 (16) Port mode registers 0, 1, 3 to 7, 12, 15**).
<5> The timer operation is enabled (TSm_n = 1).

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

6.4.3 Cautions on channel output operation

(1) Changing values set in registers TOm, TOEm, TOLm, and TOMm during timer operation

Since the timer operations (operations of TCRmn and TDRmn) are independent of the TOmn output circuit and changing the values set in TOm, TOEm, TOLm, and TOMm does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TOmn pin by timer operation, however, set TOm, TOEm, TOLm, and TOMm to the values stated in the register setting example of each operation.

When the values set in TOEm, TOLm, and TOMm (except for TOm) are changed close to the timer interrupt (INTTMmn), the waveform output to the TOmn pin may be different depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMmn) signal generation timing.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

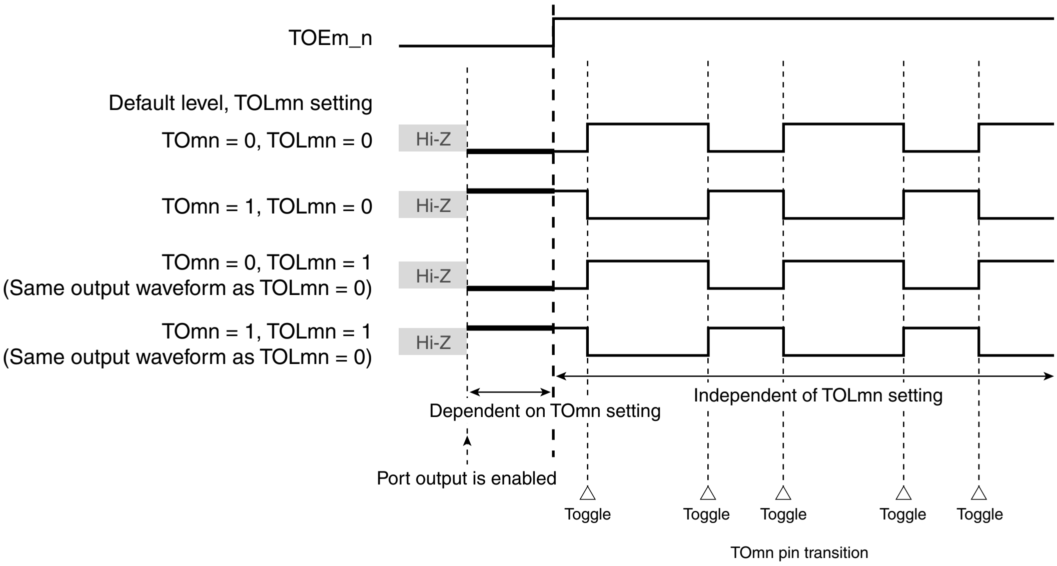
(2) Default level of TOmn pin and output level after timer operation start

The following figure shows the TOmn pin output level transition when writing has been done in the state of TOEm_n = 0 before port output is enabled and TOEm_n = 1 is set after changing the default level.

(a) When operation starts with TOMmn = 0 setting (toggle output)

The setting of TOLmn is invalid when TOMmn = 0. When the timer operation starts after setting the default level, the toggle signal is generated and the output level of TOmn pin is reversed.

Figure 6-35. TOmn Pin Output Status at Toggle Output (TOMmn = 0)



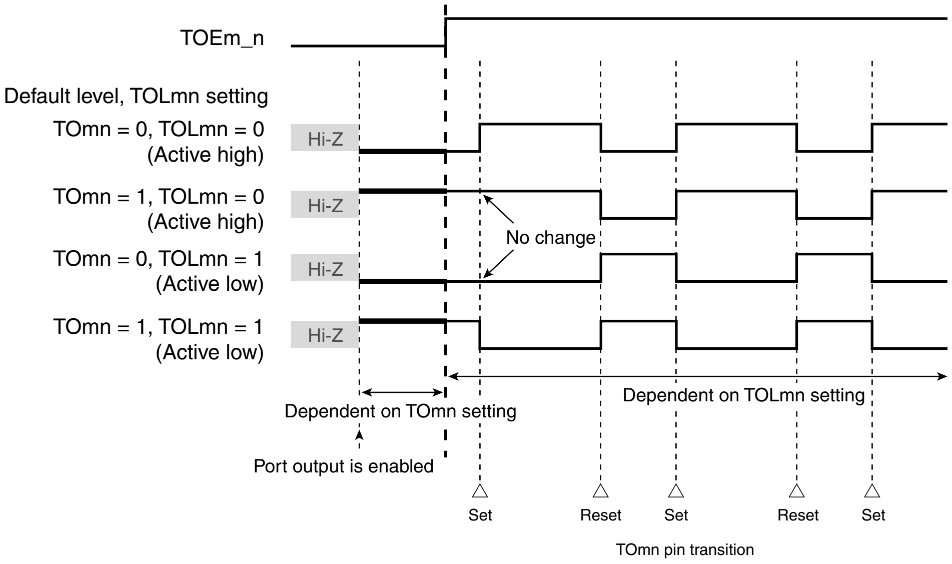
Remarks 1. Toggle: Reverse TOmn pin output status

2. m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(b) When operation starts with TOMmn = 1 setting (Combination operation mode (PWM output))

When TOMmn = 1, the active level is determined by TOLmn setting.

Figure 6-36. TOmn Pin Output Status at PWM Output (TOMmn = 1)



Remarks 1. Set: The output signal of TOmn pin changes from inactive level to active level.

Reset: The output signal of TOmn pin changes from active level to inactive level.

2. m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)

n: Channel number (n = 0 to 7)

mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3, mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

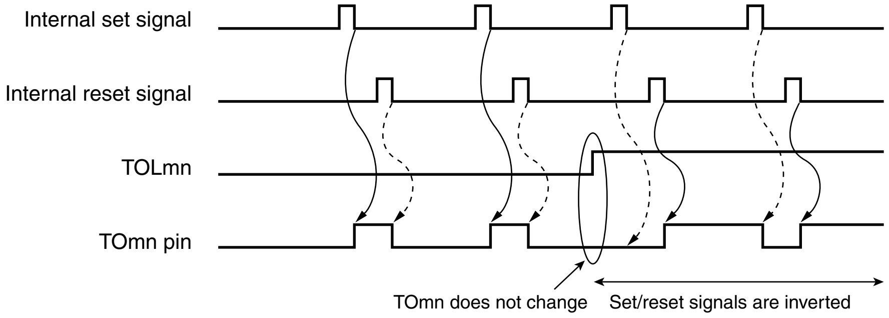
(3) Operation of TOmn pin in combination operation mode (TOMmn = 1)

(a) When TOLmn setting has been changed during timer operation

When the TOLmn setting has been changed during timer operation, the setting becomes valid at the generation timing of TOmn change condition. Rewriting TOLmn does not change the output level of TOmn.

The following figure shows the operation when the value of TOLmn has been changed during timer operation (TOMmn = 1).

Figure 6-37. Operation When TOLmn Has Been Changed During Timer Operation



Remarks 1. Set: The output signal of TOmn pin changes from inactive level to active level.
Reset: The output signal of TOmn pin changes from active level to inactive level.

2. m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(b) Set/reset timing

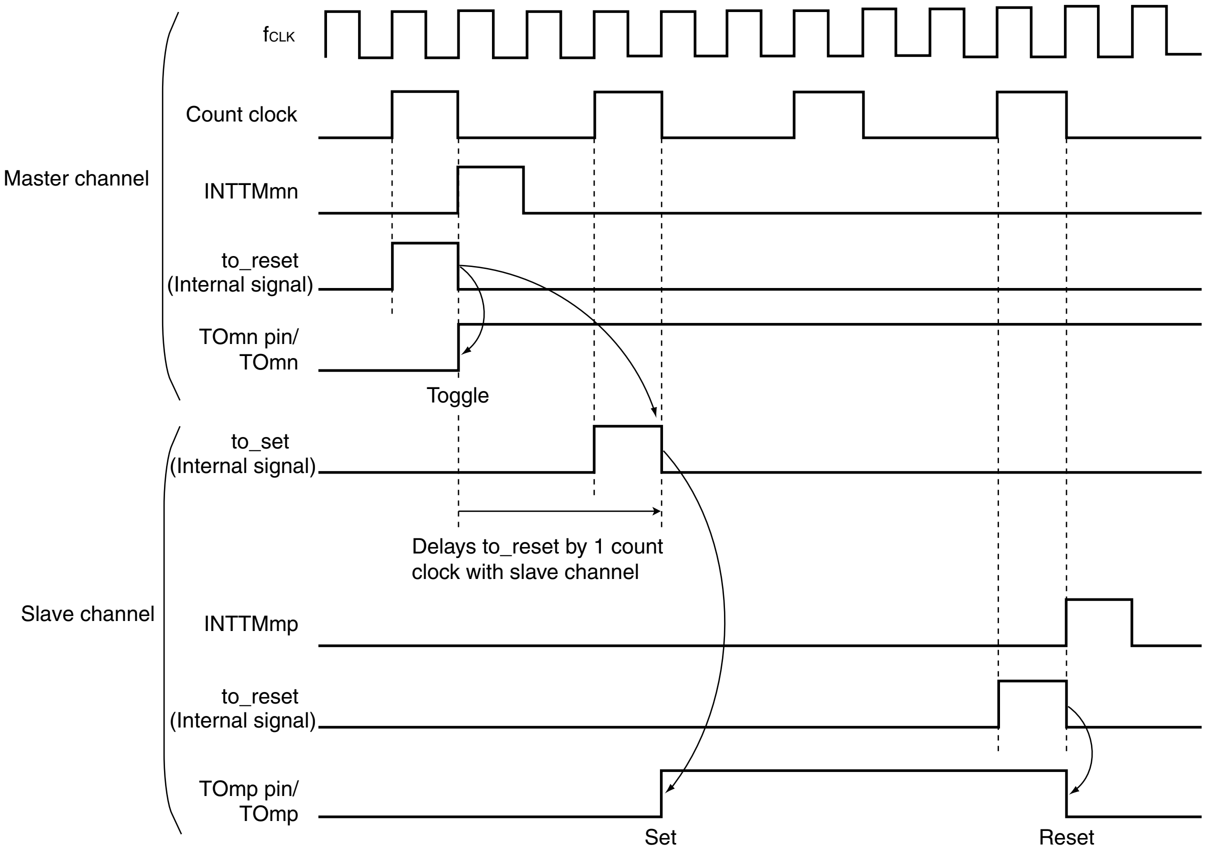
To realize 0%/100% output at PWM output, the TOmn pin/TOmn set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-38 shows the set/reset operating statuses where the master/slave channels are set as follows.

Master channel:	TOEm_n = 1, TOMmn = 0, TOLmn = 0
Slave channel:	TOEm_p = 1, TOMmp = 1, TOLmp = 0

Figure 6-38. Set/Reset Timing Operating Statuses



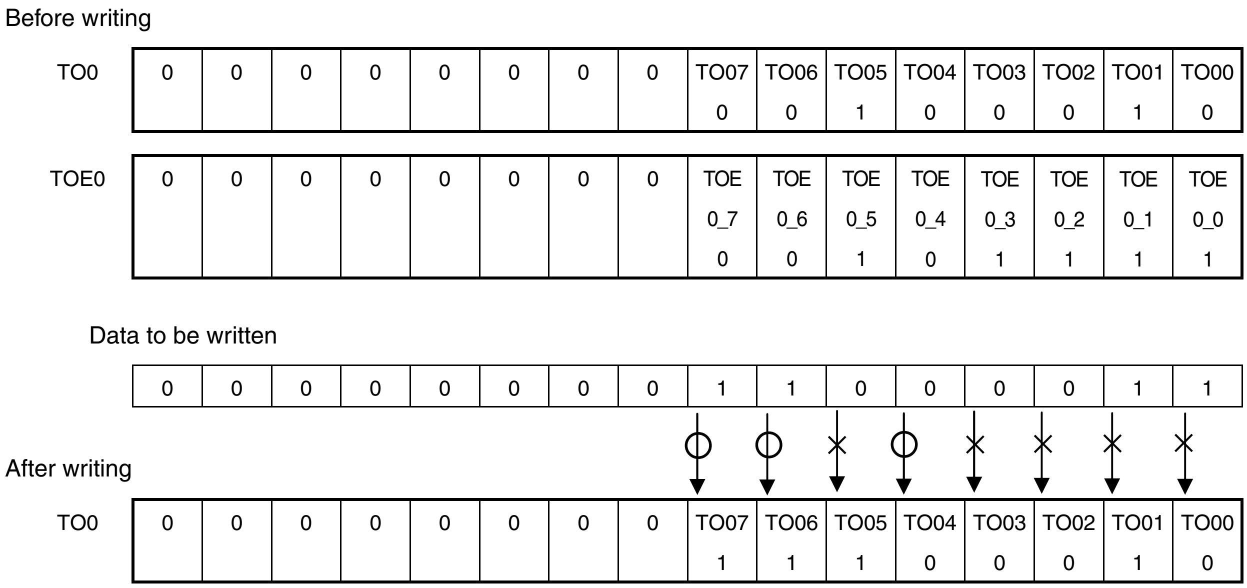
Remarks 1. to_reset: TOmn pin reset/toggle signal
to_set: TOmn pin set signal

2. m: Unit number, n: Channel number, p: Slave channel number
When m = 0
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
$n < p \leq 7$ (where p is a consecutive integer greater than n)
When m = 1
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
$n < p \leq 7$ (where p is a consecutive integer greater than n)
When m = 2
n = 0 to 3 (n = 0, 2 for master channel): 78K0R/HE3, 78K0R/HF3
$n < p \leq 3$ (where p is a consecutive integer greater than n)
n = 0 to 7 (n = 0, 2, 4, 6 for master channel): 78K0R/HG3
$n < p \leq 7$ (where p is a consecutive integer greater than n)

6.4.4 Collective manipulation of TOmn bits

In the TOm register, the setting bits (TOmn) for all the channels are located in one register in the same way as the TSm register (channel start trigger). Therefore, TOmn of all the channels can be manipulated collectively. Only specific bits can also be manipulated by setting the corresponding TOEm_n = 0 to a target TOmn (channel output).

Figure 6-39. Example of TO0n Bits Collective Manipulation



Before writing

TO0	0	0	0	0	0	0	0	0	TO07 0	TO06 0	TO05 1	TO04 0	TO03 0	TO02 0	TO01 1	TO00 0
TOE0	0	0	0	0	0	0	0	0	TOE 0_7 0	TOE 0_6 0	TOE 0_5 1	TOE 0_4 0	TOE 0_3 1	TOE 0_2 1	TOE 0_1 1	TOE 0_0 1

Data to be written

0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1
								○	○	×	○	×	×	×	×

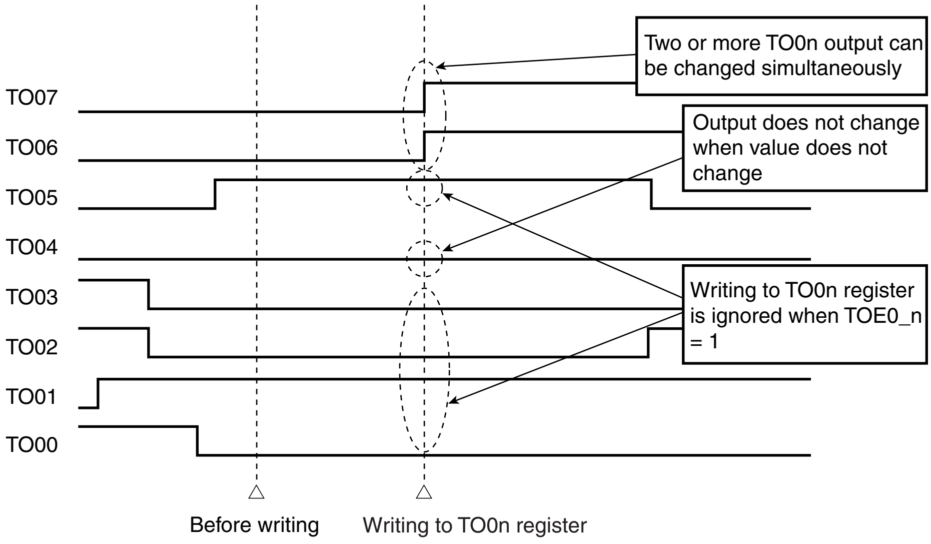
After writing

TO0	0	0	0	0	0	0	0	0	TO07 1	TO06 1	TO05 1	TO04 0	TO03 0	TO02 0	TO01 1	TO00 0

Writing is done only to TOmn bits with TOEm_n = 0, and writing to TOmn bits with TOEm_n = 1 is ignored.

TOmn (channel output) to which TOEm_n = 1 is set is not affected by the write operation. Even if the write operation is done to TOmn, it is ignored and the output change by timer operation is normally done.

Figure 6-40. TOmn Pin Statuses by Collective Manipulation of TO0n Bits


TO07
TO06
TO05
TO04
TO03
TO02
TO01
TO00
Two or more TO0n output can be changed simultaneously
Output does not change when value does not change
Writing to TO0n register is ignored when TOE0_n = 1
Before writing
Writing to TO0n register

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

Caution **When TOEm_n = 1, even if the output by timer interrupt of each channel (INTTMmn) contends with writing to TOmn, output is normally done to TOmn pin.**

Remark m Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

6.4.5 Timer interrupt and TOmn pin output at count operation start

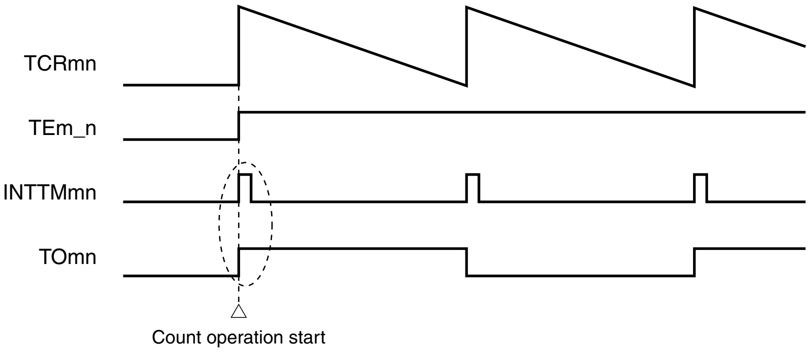
In the interval timer mode or capture mode, the MDmn0 bit in the TMRmn register sets whether or not to generate a timer interrupt at count start.

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 TOmn output is controlled.

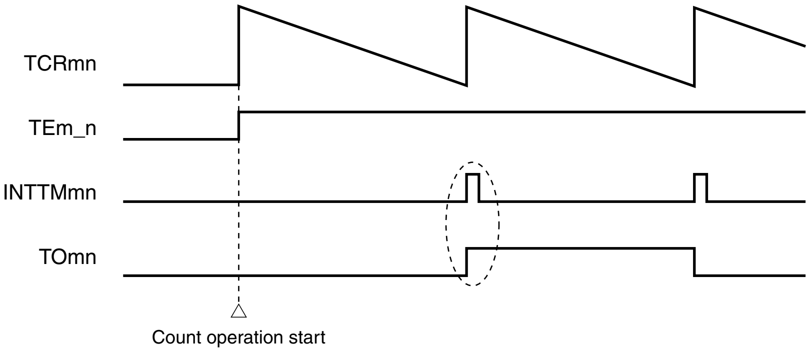
Figures 6-41 and 6-42 show operation examples when the interval timer mode (TOEm_n = 1, TOMmn = 0) is set.

Figure 6-41. When MDmn0 Is Set to 1



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

Figure 6-42. When MDmn0 Is Set to 0



When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOmn does not change either. After counting one cycle, INTTMmn is output and TOmn performs a toggle operation.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

6.5 Channel Input (TImn Pin) Control

6.5.1 TImn edge detection circuit

(1) Edge detection basic operation timing

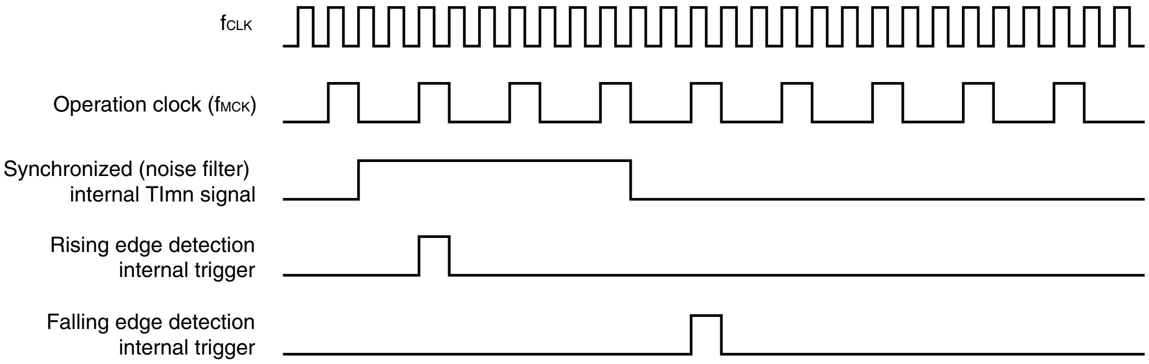
Edge detection circuit sampling is done in accordance with the operation clock (f_{MCK}).

<R> The inside signal of edge detection detects the valid edge of the input signal from a TImn pin, and detects the signal in sync with the rising edge of the next f_{MCK}.

This signal is delayed for the input signal from an actual TImn terminal by 1 to 2 clocks of f_{MCK}.

(Moreover, this signal is delayed by 3 to 4 clocks of f_{MCK} as using a noise filter.)

Figure 6-43. Edge Detection Basic Operation Timing


f_{CLK}
Operation clock (f_{MCK})
Synchronized (noise filter) internal TImn signal
Rising edge detection internal trigger
Falling edge detection internal trigger

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

6.6 Basic Function of Timer Array Unit

6.6.1 Overview of single-operation function and combination operation function

The timer array unit (TAU) 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.

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 0, 2, 4, etc.) can be set as a master channel.

(2) Any channel, except channel 0, can be set as a slave channel.

(3) The slave channel must be lower than the master channel.

Example: If channel 2 of the TAU0 is set as a master channel, channel 3 or those that follow (channels 3, 4, 5, etc.) can be set as a slave channel.

If channel 2 of the TAU2 is set as a master channel for the 78K0R/HE3 or 78K0R/HF3, channel 3 (only up to channel 3 on TAU2 of the 78K0R/HE3 or 78K0R/HF3) can be set as a slave channel.

(4) Two or more slave channels can be set for one master channel. A slave channel, however, cannot be set across a unit.

(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.

Example: If channels 0 and 4 of the TAU0 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.

(6) The operation clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKS1 and CKS0 bits (bits 15 and 14 of the TMRmn register) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.

(7) A master channel can transmit INTTMmn (interrupt), start trigger, and count clock to the lower channels.

(8) A slave channel can use the INTTMmn (interrupt), start trigger, and count clock of the master channel, but it cannot transmit its own INTTMmn (interrupt), start trigger, and count clock to the lower channel.

(9) A master channel cannot use the INTTMmn (interrupt), start trigger, and count clock from the other master channel.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

(10) To simultaneously start channels that operate in combination, the TSm_n bit of the channels in combination must be set at the same time.

(11) During a counting operation, the TSm_n bit of all channels that operate in combination or only the master channel can be set. TSm_n of only a slave channel cannot be set.

(12) To stop the channels in combination simultaneously, the TTm_n bit of the channels in combination must be set at the same time.

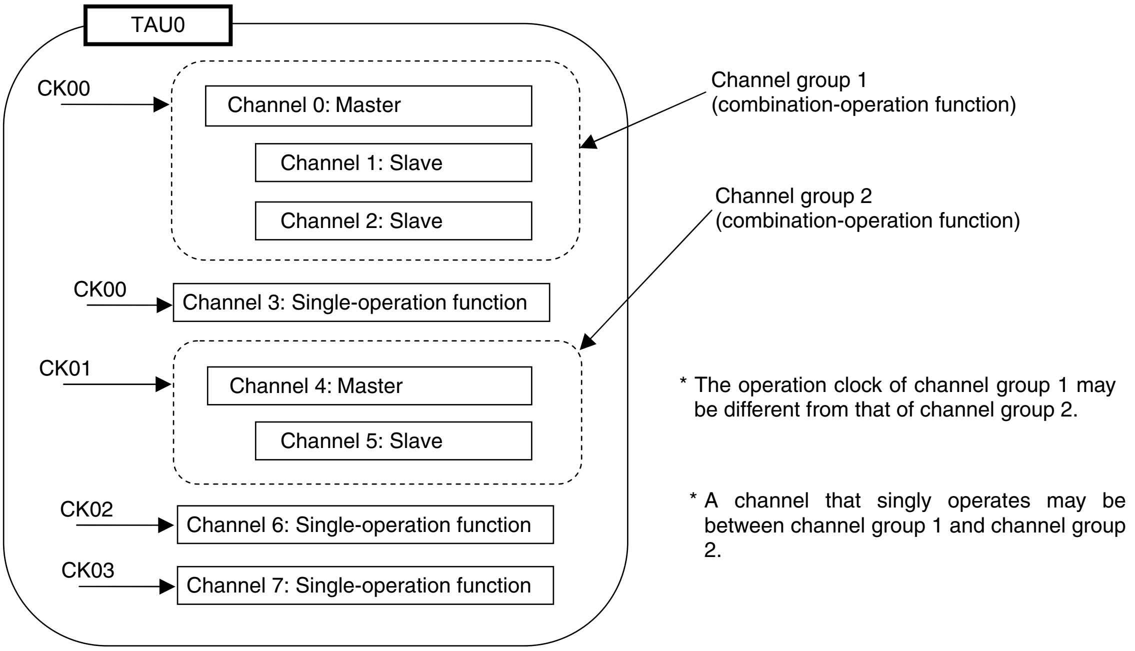
Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

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


TAU0
CK00
Channel 0: Master
Channel 1: Slave
Channel 2: Slave
CK00
Channel 3: Single-operation function
CK01
Channel 4: Master
Channel 5: Slave
CK02
Channel 6: Single-operation function
CK03
Channel 7: Single-operation function
Channel group 1 (combination-operation function)
Channel group 2 (combination-operation function)
* The operation clock of channel group 1 may be different from that of channel group 2.
* A channel that singly operates may be between channel group 1 and channel group 2.

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

TOmn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOmn can be calculated by the following expressions.

- Period of square wave output from TOmn = Period of count clock × (Set value of TDRmn + 1) × 2
- Frequency of square wave output from TOmn = Frequency of count clock/{(Set value of TDRmn + 1) × 2}

TCRmn operates as a down counter in the interval timer mode.

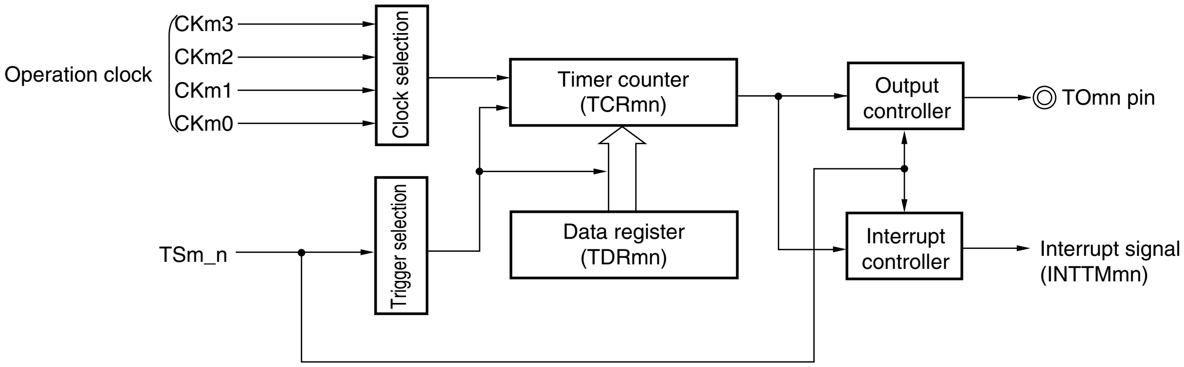
TCRmn loads the value of TDRmn at the first count clock after the channel start trigger bit (TSm_n) is set to 1. If MDmn0 of TMRmn = 0 at this time, INTTMmn is not output and TOmn is not toggled. If MDmn0 of TMRmn = 1, INTTMmn is output and TOmn is toggled.

After that, TCRmn counts down in synchronization with the count clock.

When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, TCRmn loads the value of TDRmn again. After that, the same operation is repeated.

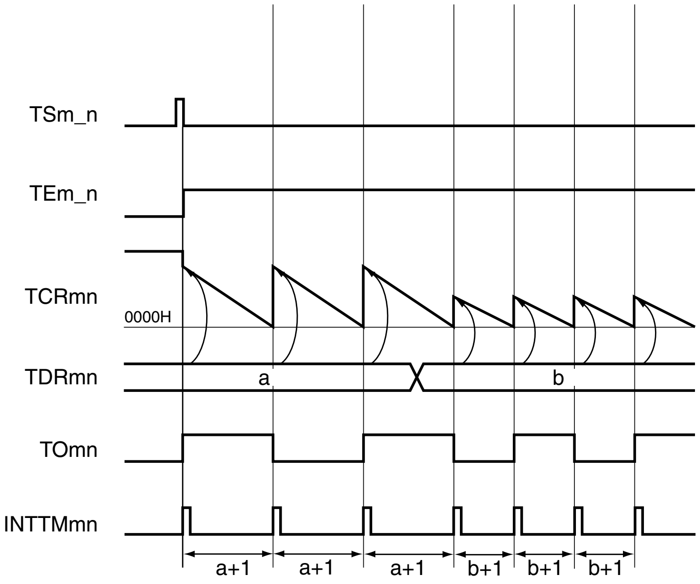
TDRmn can be rewritten at any time. The new value of TDRmn becomes valid from the next period.

Figure 6-44. Block Diagram of Operation as Interval Timer/Square Wave Output



Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

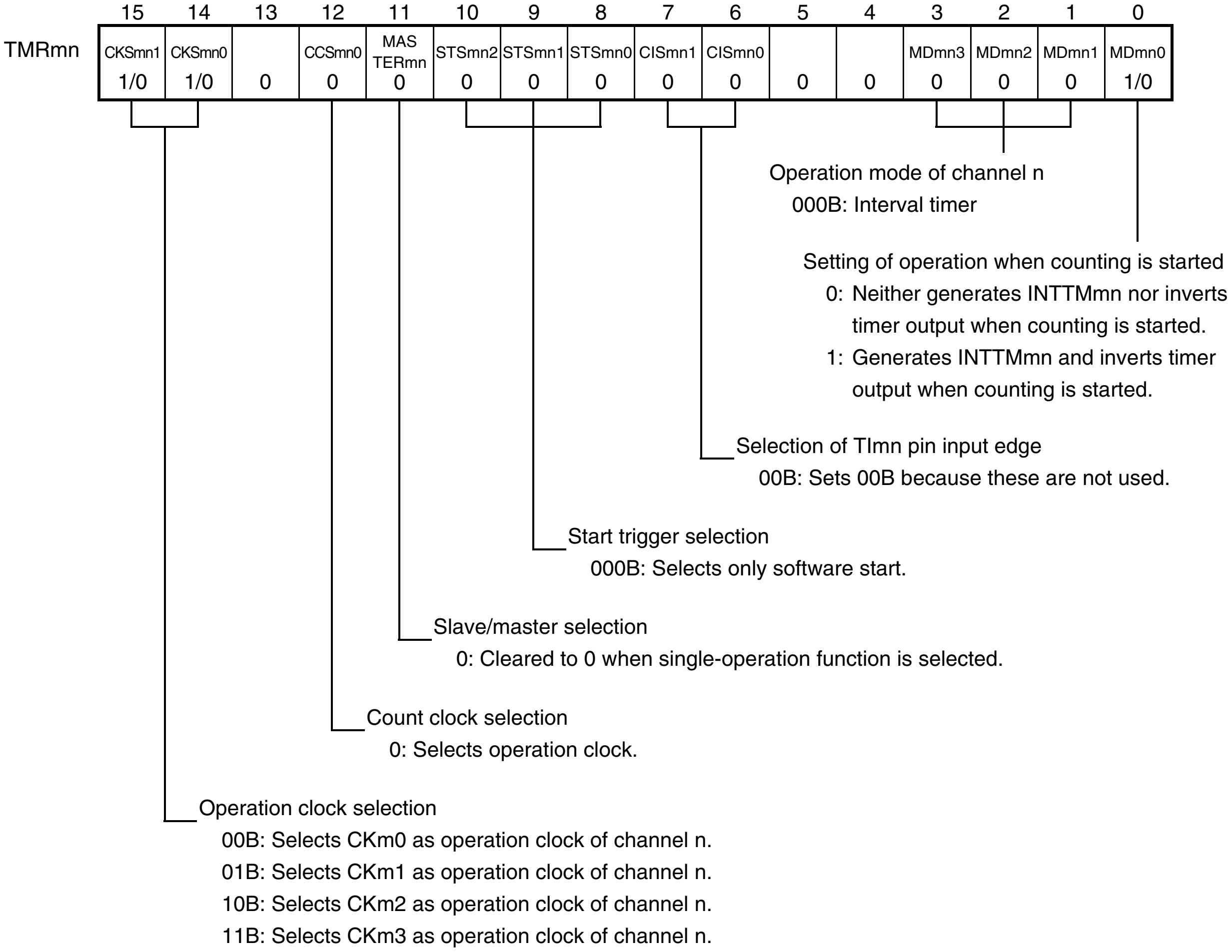
Figure 6-45. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)



Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-46. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn0	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0	0	0	0	0	0	0	0	0	0	0	1/0

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.
1: Generates INTTMmn and inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
0: Selects operation clock.

Operation clock selection
00B: Selects CKm0 as operation clock of channel n.
01B: Selects CKm1 as operation clock of channel n.
10B: Selects CKm2 as operation clock of channel n.
11B: Selects CKm3 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	1/0

0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEm_n
	1/0

0: Stops the TOmn output operation by counting operation.
1: Enables the TOmn output operation by counting operation.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-46. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/2)

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (toggle mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets toggle mode.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-47. 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.)
	The TAUmEN bit of PER0 register is set as 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 to CKm3.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel). Sets interval (period) value to the TDRmn register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOmn output Clears the TOMmn bit of the TOMm register to 0 (toggle mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output.	The TOmn pin goes into Hi-Z output state. The TOmn default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets TOEm_n to 1 and enables operation of TOmn.	TOmn does not change because channel stops operating.
	Clears the port register and port mode register to 0.	The TOmn pin outputs the TOmn set level.
Operation start	Sets TOEm_n to 1 (only when operation is resumed). Sets the TSm_n bit to 1. The TSm_n bit automatically returns to 0 because it is a trigger bit.	TEm_n = 1, and count operation starts. Value of TDRmn is loaded to TCRmn at the count clock input. INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set values of TMRmn register, TOMmn, and TOLmn bits cannot be changed. Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOm and TOEm registers can be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of 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 TTm_n bit is set to 1. The TTm_n bit automatically returns to 0 because it is a trigger bit.	TEm_n = 0, and count operation stops. TCRmn holds count value and stops. The TOmn output is not initialized but holds current status.
	TOEm_n is cleared to 0 and value is set to TOmn bit.	The TOmn pin outputs the TOmn set level.

Operation is resumed.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-47. Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOmn pin output level Clears TOmn bit to 0 after the value to be held is set to the port register.	The TOmn pin output level is held by port function.
	When holding the TOmn pin output level is not necessary Switches the port mode register to input mode.	The TOmn pin output level goes into Hi-Z output state.
	The TAUmEN bit of PER0 register is set as 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode.)

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

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 TImn pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDRmn + 1

TCRmn operates as a down counter in the event counter mode.

When the channel start trigger bit (TSm_n) is set to 1, TCRmn loads the value of TDRmn.

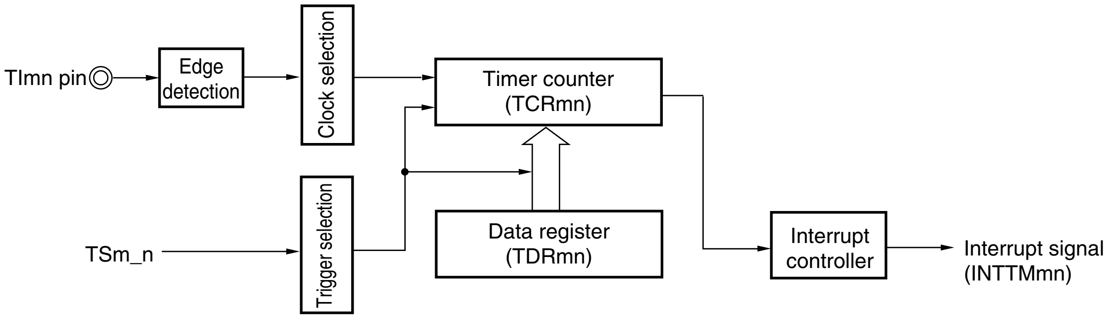
TCRmn counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, TCRmn loads the value of TDRmn again, and outputs INTTMmn.

After that, the above operation is repeated.

TOmn must not be used because its waveform depends on the external event and irregular.

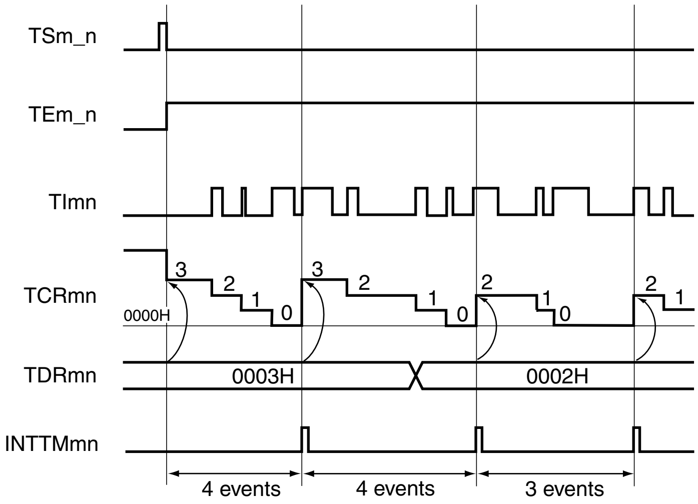
TDRmn can be rewritten at any time. The new value of TDRmn becomes valid during the next count period.

Figure 6-48. Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-49. Example of Basic Timing of Operation as External Event Counter



Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-50. Example of Set Contents of Registers in External Event Counter Mode (1/2)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	1	0	0	0	0	1/0	1/0	0	0	0	1	1	0

Operation mode of channel n
011B: Event count mode

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
1: Selects the TImn pin input valid edge.

Operation clock selection
00B: Selects CKm0 as operation clock of channel n.
01B: Selects CKm1 as operation clock of channel n.
10B: Selects CKm2 as operation clock of channel n.
11B: Selects CKm3 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEm_n
	0

0: Stops the TOmn output operation by counting operation.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-50. Example of Set Contents of Registers in External Event Counter Mode (2/2)

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (toggle mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets toggle mode.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-51. 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.)
	The TAUmEN bit of PER0 register is set as 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 to CKm3.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel). Sets number of counts to the TDRmn register. Clears the TOEm_n bit of the TOEm register to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSm_n bit to 1. → The TSm_n bit automatically returns to 0 because it is a trigger bit.	TEm_n = 1, and count operation starts. Value of TDRmn is loaded to TCRmn and detection of the TImn pin input edge is awaited.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEm_n bits cannot be changed.	Counter (TCRmn) counts down each time input edge of the TImn pin has been detected. When count value reaches 0000H, the value of TDRmn is loaded to TCRmn again, and the count operation is continued. By detecting TCRmn = 0000H, the INTTMmn output is generated. After that, the above operation is repeated.
Operation stop	The TTm_n bit is set to 1. → The TTm_n bit automatically returns to 0 because it is a trigger bit.	TEm_n = 0, and count operation stops. TCRmn holds count value and stops.
TAU stop	The TAUmEN bit of PER0 register is set as 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

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 TImn pin and outputs the result from TOmn.

The divided clock frequency output from TOmn can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDRmn + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDRmn + 1)

TCRmn operates as a down counter in the interval timer mode.

After the channel start trigger bit (TSm_n) is set to 1, TCRmn loads the value of TDRmn when the TImn valid edge is detected. If MDmn0 of TMRmn = 0 at this time, INTTMmn is not output and TOmn is not toggled. If MDmn0 of TMRmn = 1, INTTMmn is output and TOmn is toggled.

After that, TCRmn counts down at the valid edge of TImn. When TCRmn = 0000H, it toggles TOmn. At the same time, TCRmn loads the value of TDRmn again, and continues counting.

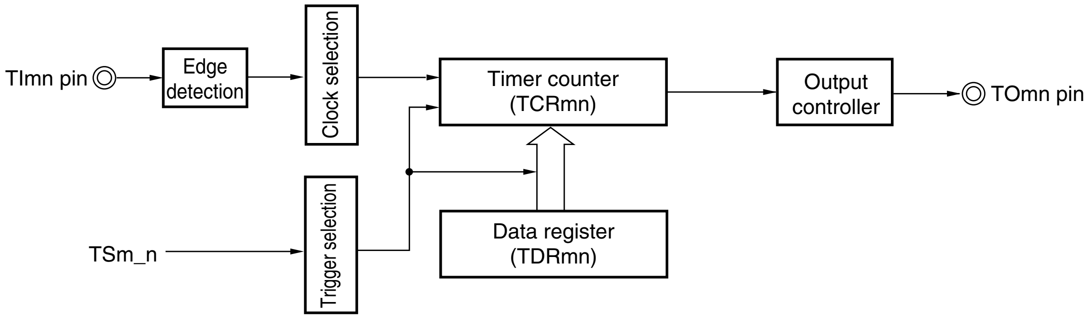
If detection of both the edges of TImn is selected, the duty factor error of the input clock affects the divided clock period of the TOmn output.

The period of the TOmn output clock includes a sampling error of one period of the operation clock.

Clock period of TOmn output = Ideal TOmn output clock period ± Operation clock period (error)

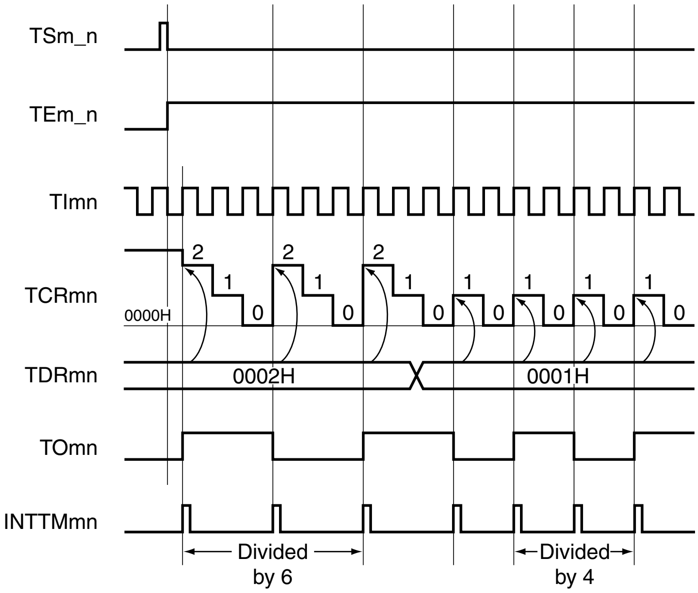
TDRmn can be rewritten at any time. The new value of TDRmn becomes valid during the next count period.

Figure 6-52. Block Diagram of Operation as Frequency Divider



Remark m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 7)
mn = 05, 11: 78K0R/HC3, 78K0R/HE3, mn = 00 to 02, 04 to 07, 10 to 17: 78K0R/HF3,
mn = 00 to 07, 10 to 17: 78K0R/HG3

Figure 6-53. Example of Basic Timing of Operation as Frequency Divider (MDmn0 = 1)



Remark m: Unit number (m = 0, 1)

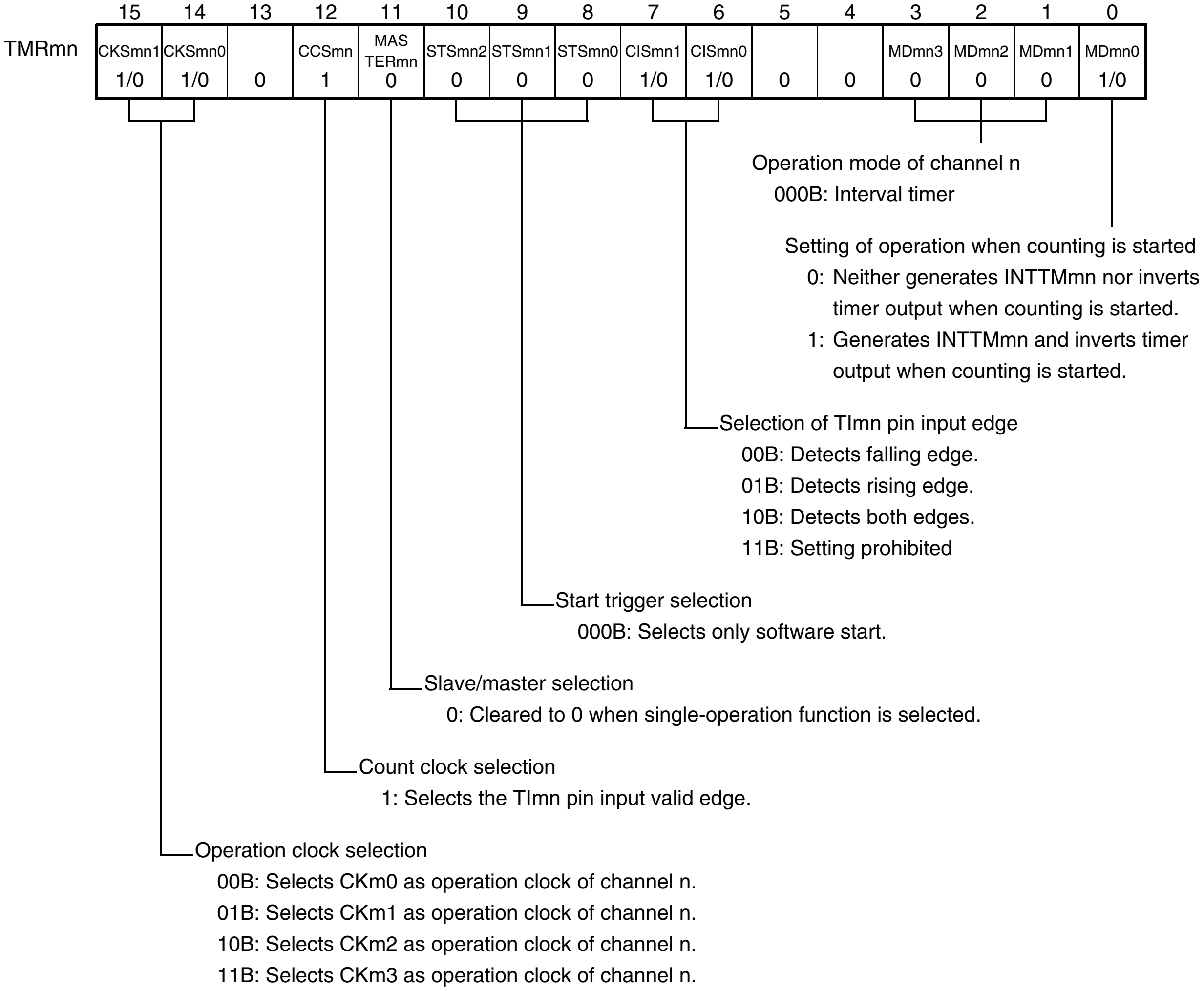
n: Channel number (n = 0 to 7)

mn = 05, 11: 78K0R/HC3, 78K0R/HE3, mn = 00 to 02, 04 to 07, 10 to 17: 78K0R/HF3,

mn = 00 to 07, 10 to 17: 78K0R/HG3

Figure 6-54. Example of Set Contents of Registers When Frequency Divider Is Used (1/2)

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	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 TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
1: Selects the TImn pin input valid edge.

Operation clock selection
00B: Selects CKm0 as operation clock of channel n.
01B: Selects CKm1 as operation clock of channel n.
10B: Selects CKm2 as operation clock of channel n.
11B: Selects CKm3 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	1/0

0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEm_n
	1/0

0: Stops the TOmn output operation by counting operation.
1: Enables the TOmn output operation by counting operation.

Remark m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 7)
mn = 05, 11: 78K0R/HC3, 78K0R/HE3, mn = 00 to 02, 04 to 07, 10 to 17: 78K0R/HF3,
mn = 00 to 07, 10 to 17: 78K0R/HG3

Figure 6-54. Example of Set Contents of Registers When Frequency Divider Is Used (2/2)

(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: Unit number (m = 0, 1)

n: Channel number (n = 0 to 7)

mn = 05, 11: 78K0R/HC3, 78K0R/HE3, mn = 00 to 02, 04 to 07, 10 to 17: 78K0R/HF3,
mn = 00 to 07, 10 to 17: 78K0R/HG3

Figure 6-55. 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.)
	The TAUmEN bit of PER0 register is set as 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 to CKm3.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel). Sets interval (period) value to the TDRmn register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOMmn bit of the TOMm register to 0 (toggle mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output. → Sets TOEm_n to 1 and enables operation of TOmn. → Clears the port register and port mode register to 0. →	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. TOmn does not change because channel stops operating. The TOmn pin outputs the TOmn set level.
Operation start	Sets the TOEm_n to 1 (only when operation is resumed). Sets the TSm_n bit to 1. → The TSm_n bit automatically returns to 0 because it is a trigger bit.	TEm_n = 1, and count operation starts. Value of TDRmn is loaded to TCRmn at the count clock input. INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of TOm and TOEm registers can be changed. Set values of TMRmn register, TOMmn, and TOLmn bits cannot 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 TTm_n bit is set to 1. → The TTm_n bit automatically returns to 0 because it is a trigger bit.	TEm_n = 0, and count operation stops. TCRmn holds count value and stops. The TOmn output is not initialized but holds current status.
	TOEm_n is cleared to 0 and value is set to the TOmn bit. →	The TOmn pin outputs the TOmn set level.
TAU stop	To hold the TOmn pin output level Clears TOmn bit to 0 after the value to be held is set to the port register. → When holding the TOmn pin output level is not necessary Switches the port mode register to input mode. →	The TOmn pin output level is held by port function. The TOmn pin output level goes into Hi-Z output state.
	The TAUmEN bit of PER0 register is set as 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode).

Operation is resumed.

Remark m: Unit number (m = 0, 1)

n: Channel number (n = 0 to 7)

mn = 05, 11: 78K0R/HC3, 78K0R/HE3, mn = 00 to 02, 04 to 07, 10 to 17: 78K0R/HF3,

mn = 00 to 07, 10 to 17: 78K0R/HG3

6.7.4 Operation as input pulse interval measurement

The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured.

The pulse interval can be calculated by the following expression.

TImn input pulse interval = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operation clock selected with the CKSmn bit of the TMRmn register, so an error at a maximum of one clock is generated.

TCRmn operates as an up counter in the capture mode.

When the channel start trigger (TSm_n) is set to 1, TCRmn counts up from 0000H in synchronization with the count clock.

When the TImn pin input valid edge is detected, the count value is transferred (captured) to TDRmn and, at the same time, the counter (TCRmn) is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of the TSRmn register is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

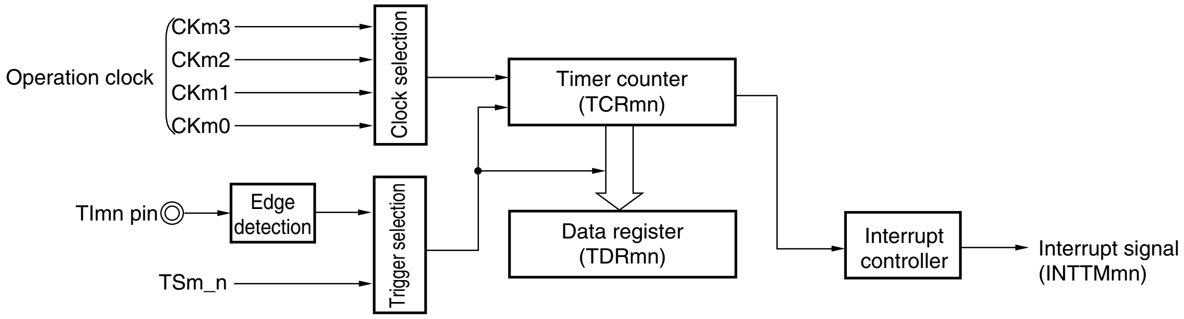
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, the OVF bit is configured as a cumulative flag, the correct interval value cannot be measured if an overflow occurs more than once.

Set STSmn2 to STSmn0 of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

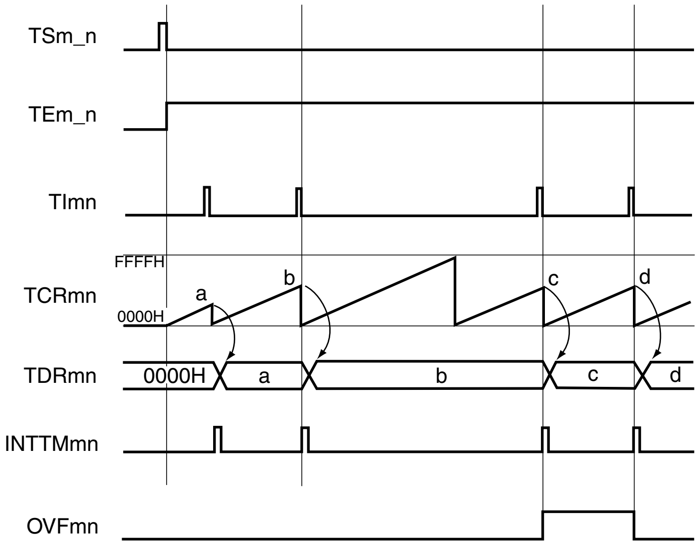
When TEm_n = 1, instead of the TImn pin input, a software operation (TSm_n = 1) can be used as a capture trigger.

Figure 6-56. Block Diagram of Operation as Input Pulse Interval Measurement



Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-57. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)



Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-58. Example of Set Contents of Registers to Measure Input Pulse Interval (1/2)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0	0	0	1	1/0	1/0	0	0	0	1	0	1/0

Operation mode of channel n
010B: Capture mode

Setting of operation when counting is started
0: Does not generate INTTMmn when counting is started.
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Capture trigger selection
001B: Selects the TImn pin input valid edge.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
0: Selects operation clock.

Operation clock selection
00B: Selects CKm0 as operation clock of channel n.
01B: Selects CKm1 as operation clock of channel n.
10B: Selects CKm2 as operation clock of channel n.
11B: Selects CKm3 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEm_n
	0

0: Stops TOmn output operation by counting operation.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-58. Example of Set Contents of Registers to Measure Input Pulse Interval (2/2)

(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: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-59. 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.)
	The TAUmEN bit of PER0 register is set as 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 to CKm3.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSm_n bit to 1. The TSm_n bit automatically returns to 0 because it is a trigger bit.	TEm_n = 1, and count operation starts. TCRmn is cleared to 0000H at the count clock input. When the MDmn0 bit of the TMRmn register is 1, INTTMmn is generated.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of TOMmn, TOLmn, TOmn, and TOEm_n bits cannot be changed.	Counter (TCRmn) counts up from 0000H. When the TImn pin input valid edge is detected, the count value is transferred (captured) to TDRmn. At the same time, TCRmn is cleared to 0000H, and the INTTMmn signal is generated. If an overflow occurs at this time, the OVF bit of the TSRmn register is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TTm_n bit is set to 1. The TTm_n bit automatically returns to 0 because it is a trigger bit.	TEm_n = 0, and count operation stops. TCRmn holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of PER0 register is set as 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

6.7.5 Operation as input signal high-/low-level width measurement

By starting counting at one edge of TImn and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TImn can be measured. The signal width of TImn can be calculated by the following expression.

Signal width of TImn input = Period of count clock × ((10000H × TSRn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operation clock selected with the CKSmn bit of the TMRmn register, so an error at a maximum of one clock is generated.

TCRmn operates as an up counter in the capture & one-count mode.

When the channel start trigger (TSm_n) is set to 1, TEm_n is set to 1 and the TImn pin start edge detection wait status is set.

When the TImn start valid edge (rising edge of TImn 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 TImn when the high-level width is to be measured) is detected later, the count value is transferred to TDRmn and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of the TSRmn register is set to 1. If the counter does not overflow, the OVF bit is cleared. TCRmn stops at the value "value transferred to TDRmn + 1", and the TImn pin start edge detection wait status is set. After that, the above operation is repeated.

As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, the OVF bit is configured as an integral flag, and the correct interval value cannot be measured if an overflow occurs more than once.

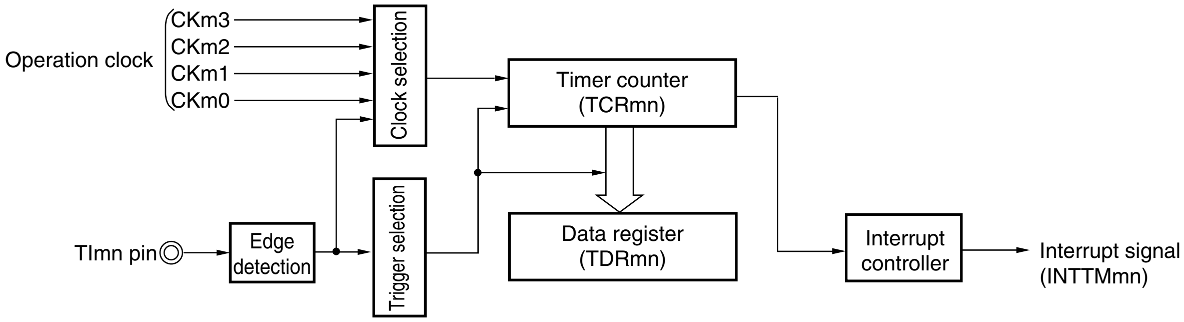
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.

Because this function is used to measure the signal width of the TImn pin input, TSm_n cannot be set to 1 while TEm_n is 1.

CISmn1, CISmn0 of TMRmn = 10B: Low-level width is measured.
CISmn1, CISmn0 of TMRmn = 11B: High-level width is measured.

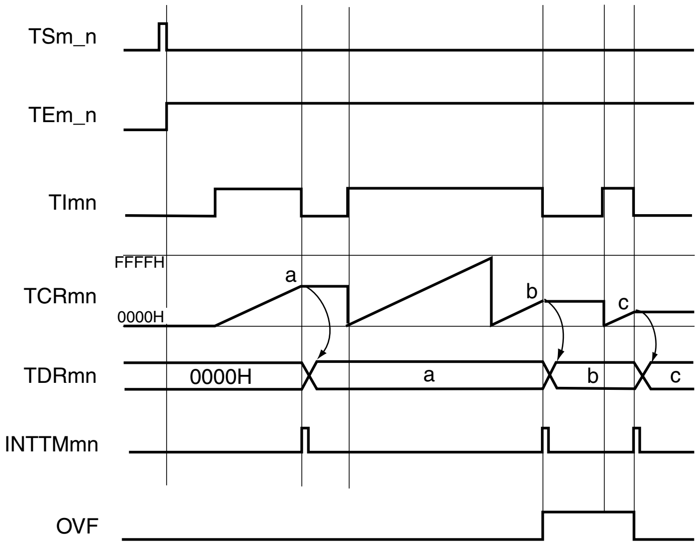
Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-60. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

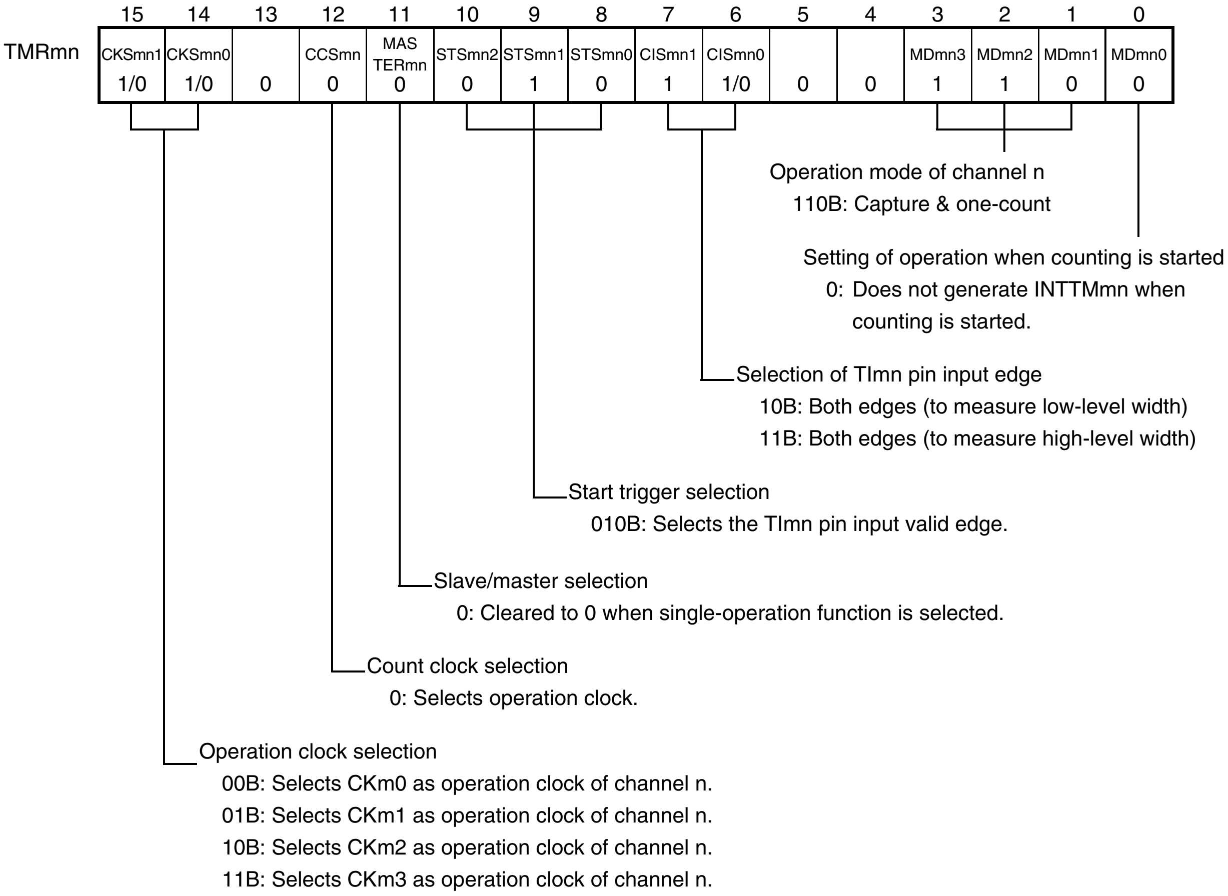
Figure 6-61. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-62. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width (1/2)

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0	0	1	0	1	1/0	0	0	1	1	0	0

Operation mode of channel n
110B: Capture & one-count

Setting of operation when counting is started
0: Does not generate INTTMmn when counting is started.

Selection of TImn pin input edge
10B: Both edges (to measure low-level width)
11B: Both edges (to measure high-level width)

Start trigger selection
010B: Selects the TImn pin input valid edge.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
0: Selects operation clock.

Operation clock selection
00B: Selects CKm0 as operation clock of channel n.
01B: Selects CKm1 as operation clock of channel n.
10B: Selects CKm2 as operation clock of channel n.
11B: Selects CKm3 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEm_n 0	0: Stops the TOmn output operation by counting operation.

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-62. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width (2/2)

(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: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 6-63. 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.)
	The TAUmEN bit of PER0 register is set as 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 to CKm3.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel). Clears TOEm_n to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSm_n bit to 1. The TSm_n bit automatically returns to 0 because it is a trigger bit.	TEm_n = 1, and the TImn pin start edge detection wait status is set.
	Detects TImn pin input count start valid edge.	Clears TCRmn to 0000H and starts counting up.
During operation	The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEm_n bits cannot be changed.	When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to TDRmn and INTTMmn is generated. If an overflow occurs at this time, the OVF bit of the TSRmn register is set; if an overflow does not occur, the OVF bit is cleared. TCRmn stops the count operation until the next TImn pin start edge is detected. After that, the above operation is repeated.
Operation stop	The TTm_n bit is set to 1. TTm_n bit automatically returns to 0 because it is a trigger bit.	TEm_n = 0, and count operation stops. TCRmn holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of PER0 register is set as 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

<R>

<R>

Remark m: Unit number (m = 0, 1: 78K0R/HC3, m = 0 to 2: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3)
n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 17: 78K0R/HC3, mn = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
mn = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

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 Although the duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it is summarized into 100% output.

The master channel operates in the interval timer mode and counts the periods. When the channel start trigger (TSm_n) 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.

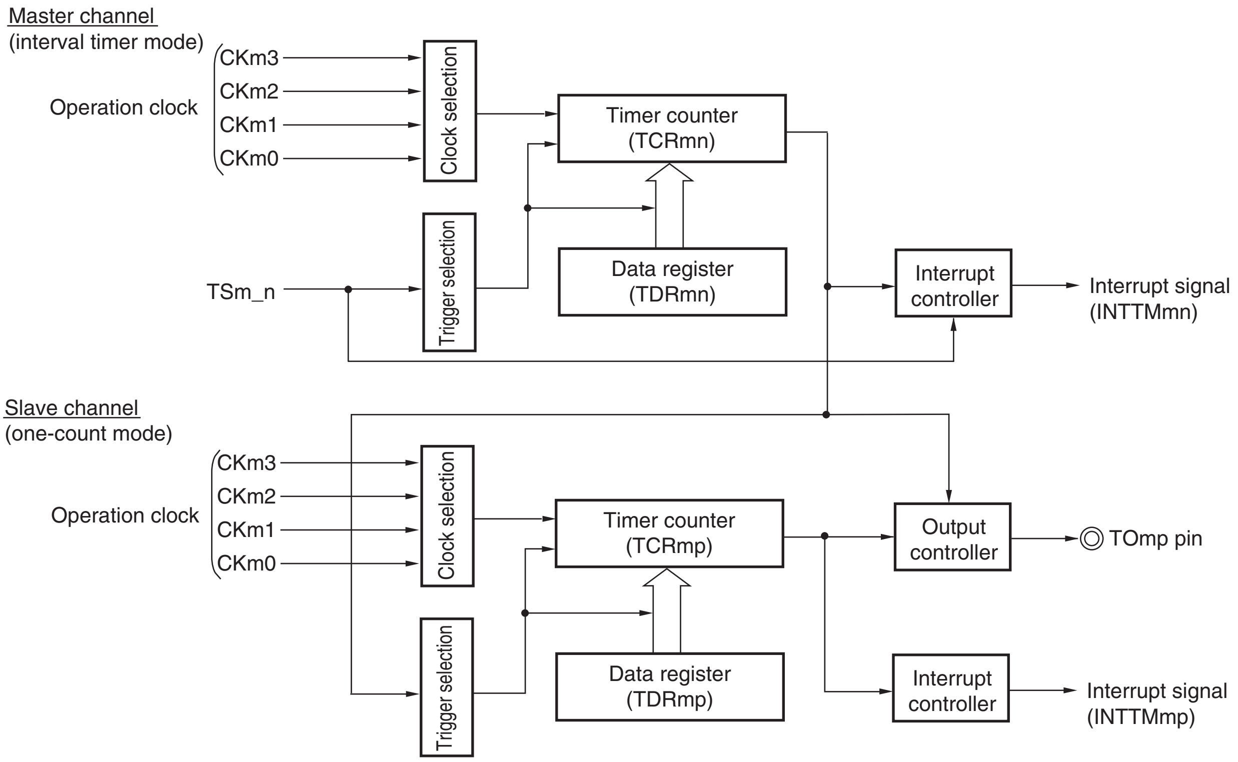
The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

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 TRCmp 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: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)
When m = 0: n = 0, 2, 4, 6
When m = 1:
n = 0, 2, 4, 6
When m = 2:
n = 0, 2: 78K0R/HE3, 78K0R/HF3
n = 0, 2, 4, 6: 78K0R/HG3

<R>

Figure 6-64. Block Diagram of Operation as PWM Function



Remark m: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)

When m = 0: n = 0, 2, 4, 6

When m = 1:

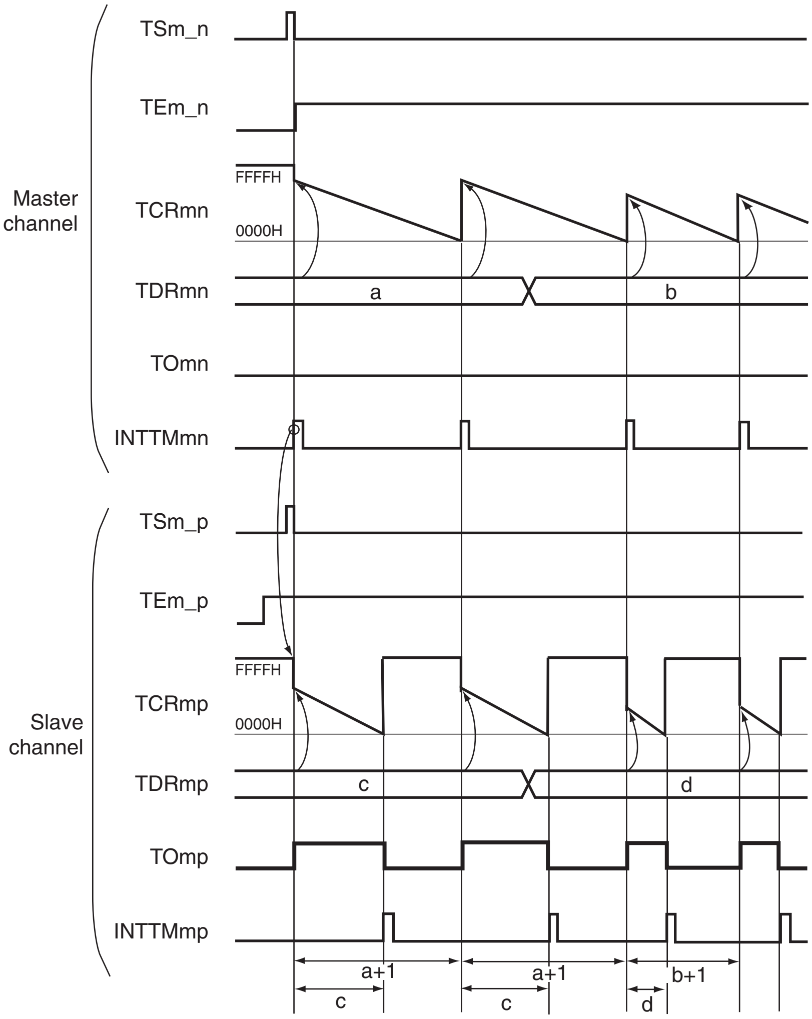
n = 0, 2, 4, 6

When m = 2:

n = 0, 2: 78K0R/HE3, 78K0R/HF3

n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-65. Example of Basic Timing of Operation as PWM Function



Remark m: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)

When m = 0: n = 0, 2, 4, 6

When m = 1:

n = 0, 2, 4, 6

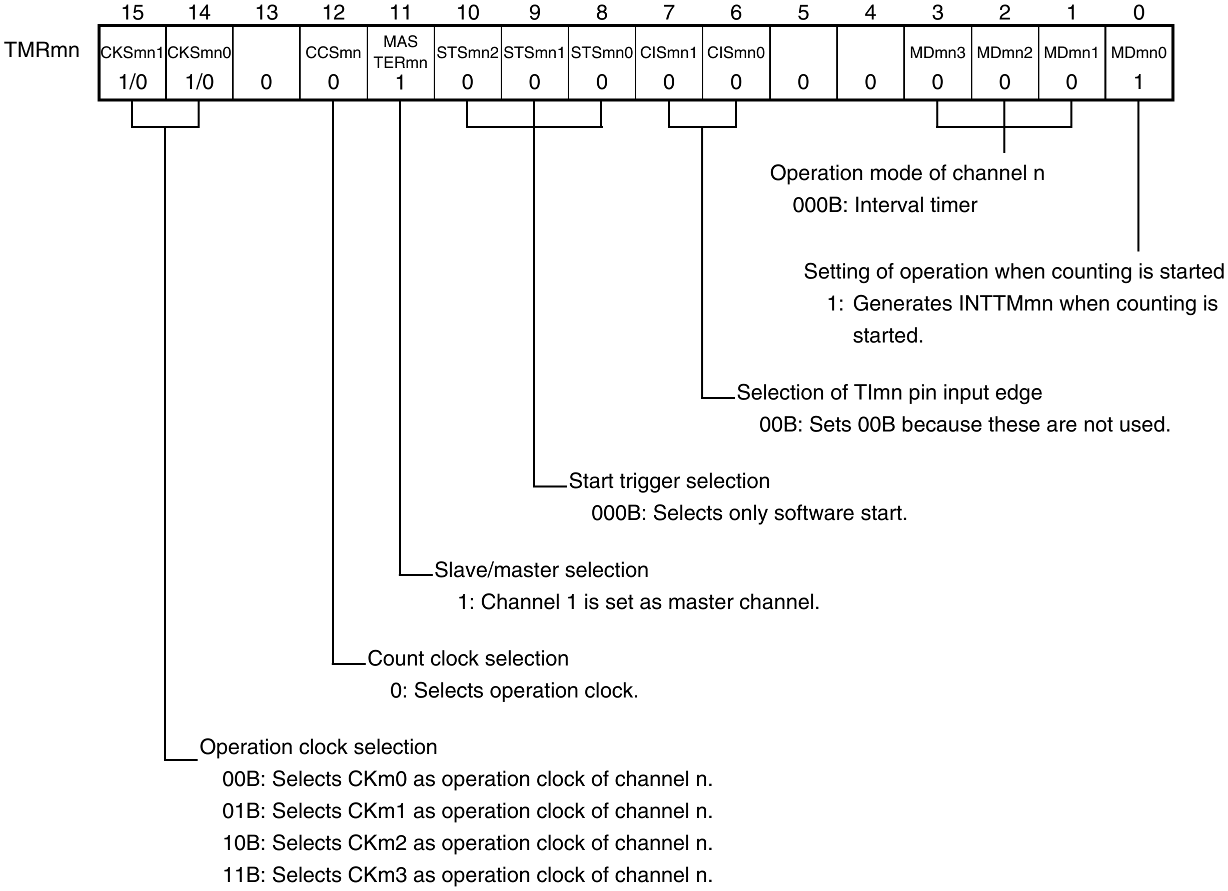
When m = 2:

n = 0, 2: 78K0R/HE3, 78K0R/HF3

n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-66. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used (1/2)

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
00B: Selects CKm0 as operation clock of channel n.
01B: Selects CKm1 as operation clock of channel n.
10B: Selects CKm2 as operation clock of channel n.
11B: Selects CKm3 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEm_n 0	0: Stops the TOmn output operation by counting operation.

Remark m: Unit number, n: Master channel number
When m = 0: n = 0, 2, 4, 6
When m = 1: n = 0, 2, 4, 6
When m = 2: n = 0, 2: 78K0R/HE3, 78K0R/HF3, n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-66. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used (2/2)

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (toggle mode).

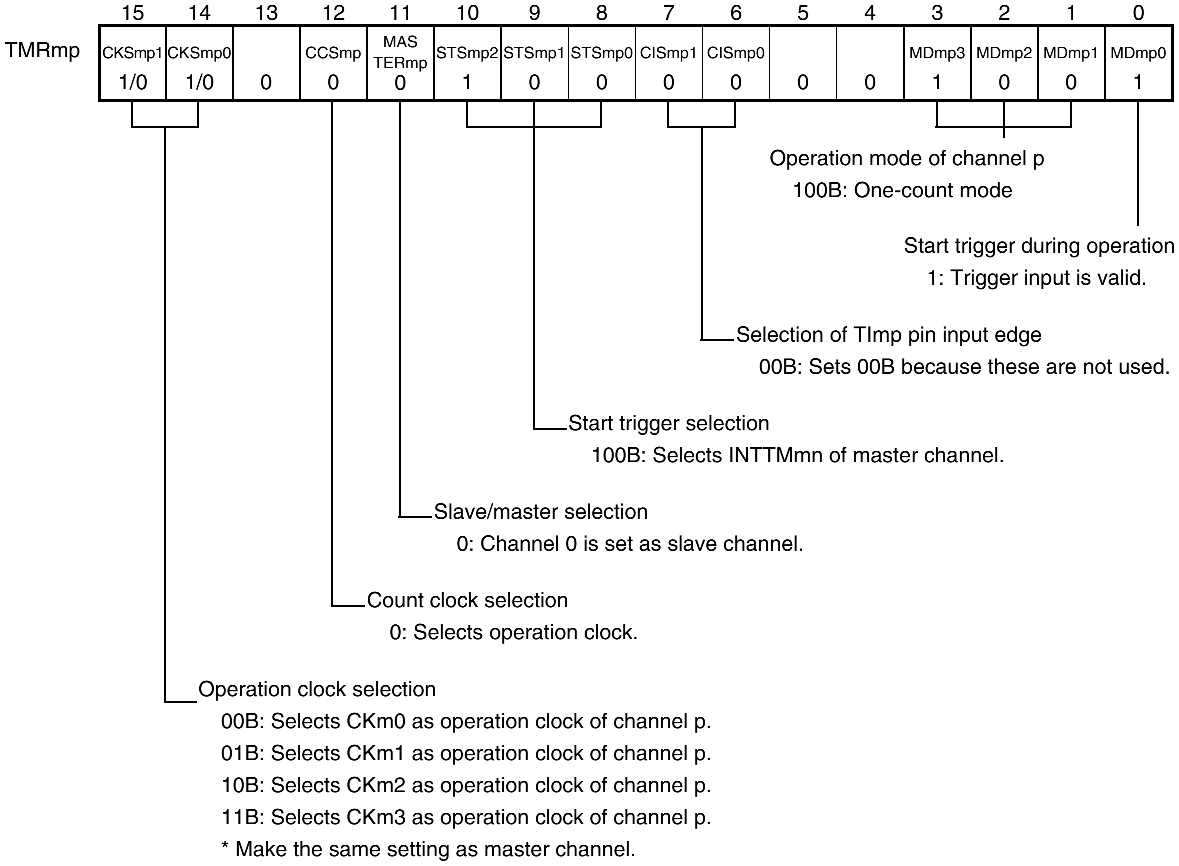
(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets toggle mode.

Remark m: Unit number, n: Master channel number
When m = 0: n = 0, 2, 4, 6
When m = 1: n = 0, 2, 4, 6
When m = 2: n = 0, 2: 78K0R/HE3, 78K0R/HF3, n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-67. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used (1/2)

(a) Timer mode register mp (TMRmp)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	MAS TERmp	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	1/0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
00B: Selects CKm0 as operation clock of channel p.
01B: Selects CKm1 as operation clock of channel p.
10B: Selects CKm2 as operation clock of channel p.
11B: Selects CKm3 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p
TOm	TOmp
	1/0

0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p
TOEm	TOEm_p
	1/0

0: Stops the TOmp output operation by counting operation.
1: Enables the TOmp output operation by counting operation.

Remark m: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)
When m = 0: n = 0, 2, 4, 6
When m = 1: n = 0, 2, 4, 6
When m = 2: n = 0, 2: 78K0R/HE3, 78K0R/HF3, n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-67. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used (2/2)

(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: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)

When m = 0: n = 0, 2, 4, 6

When m = 1: n = 0, 2, 4, 6

When m = 2: n = 0, 2: 78K0R/HE3, 78K0R/HF3, n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-68. 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.)
	The TAUmEN bit of PER0 register is set as 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 to CKm3.	
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.	The TOmp pin goes into Hi-Z output state.
	Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEm_p 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.

Remark m: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)

When m = 0: n = 0, 2, 4, 6

When m = 1:

n = 0, 2, 4, 6

When m = 2:

n = 0, 2: 78K0R/HE3, 78K0R/HF3

n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-68. Operation Procedure When PWM Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets TOEm_p (slave) to 1 (only when operation is resumed). The TSm_n (master) and TSm_p (slave) bits of the TSm register are set to 1 at the same time. The TSm_n and TSm_p bits automatically return to 0 because they are trigger bits.	TEm_n = 1, TEm_p = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation <R> <R>	Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, TOmn, and TOEmn 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 TOLmp, TOmp, and TOEmp bits can 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 TTm_n (master) and TTm_p (slave) bits are set to 1 at the same time. The TTm_n and TTm_p bits automatically return to 0 because they are trigger bits.	TEm_n, TEm_p = 0, and count operation stops. TCRmn and TCRmp hold count value and stops. The TOmp output is not initialized but holds current status.
	TOEm_p of slave channel is cleared to 0 and value is set to the TOmp bit.	The TOmp pin outputs the TOmn set level.
TAU stop	To hold the TOmp pin output level Clears TOmp bit to 0 after the value to be held is set to the port register. When holding the TOmp pin output level is not necessary Switches the port mode register to input mode.	The TOmp pin output level is held by port function. The TOmp pin output level goes into Hi-Z output state.
	The TAUmEN bit of PER0 register is set as 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)

When m = 0: n = 0, 2, 4, 6

When m = 1:

n = 0, 2, 4, 6

When m = 2:

n = 0, 2: 78K0R/HE3, 78K0R/HF3

n = 0, 2, 4, 6: 78K0R/HG3

6.8.2 Operation as one-shot pulse output function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the signal input to the TImn pin.

The delay time and pulse width can be calculated by the following expressions.

> Delay time = {Set value of TDRmn (master) + 2} × Count clock period
> Pulse width = {Set value of TDRmp (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. 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.

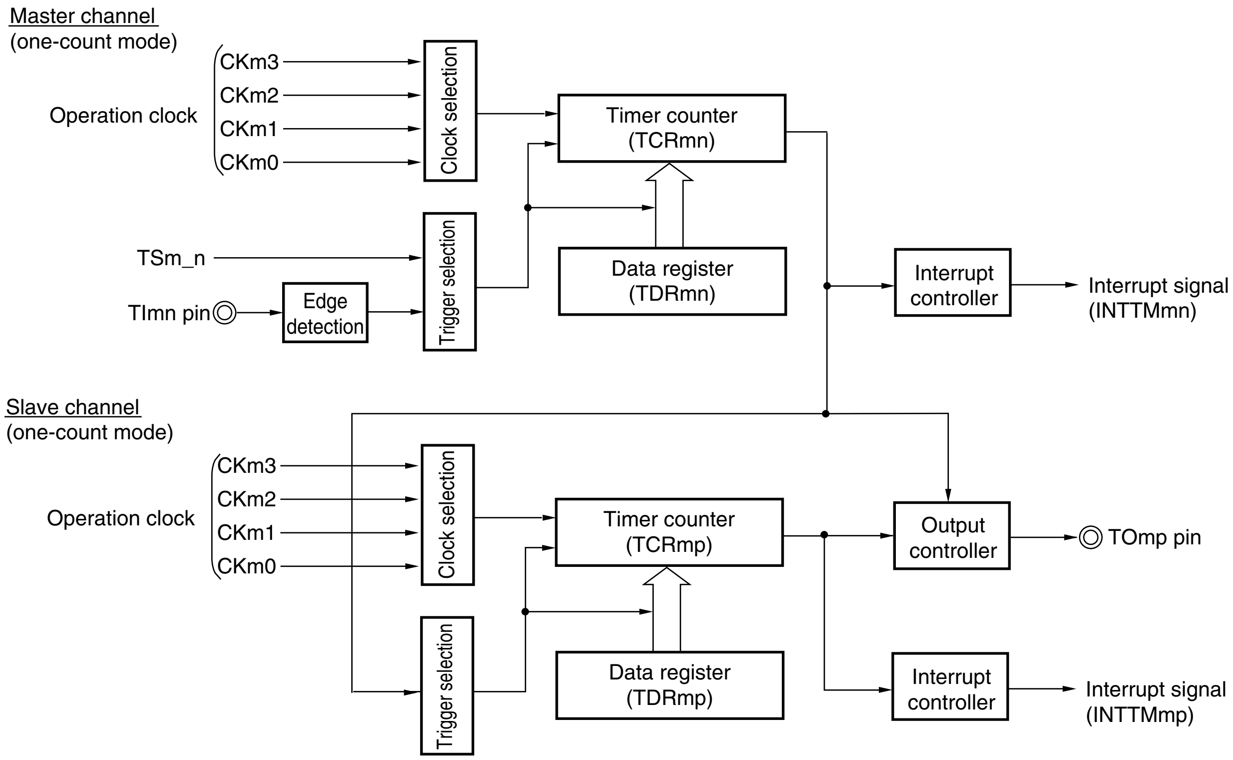
The slave channel operates in the one-count mode and counts the pulse width. TCRmp of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the TDRmp value. TCRmp counts down from the value of TDRmp it has loaded, in synchronization with the count value. 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 (TSm_n = 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. Rewrite the TDRmn immediately after INTTMmn is generated and the TDRmp immediately after INTTMmp is generated.**

Remark m: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)
When m = 0: n = 0, 2, 4, 6
When m = 1:
n = 0, 2, 4, 6
When m = 2:
n = 0, 2: 78K0R/HE3, 78K0R/HF3
n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-69. Block Diagram of Operation as One-Shot Pulse Output Function



Remark m: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)

When m = 0: n = 0, 2, 4, 6

When m = 1:

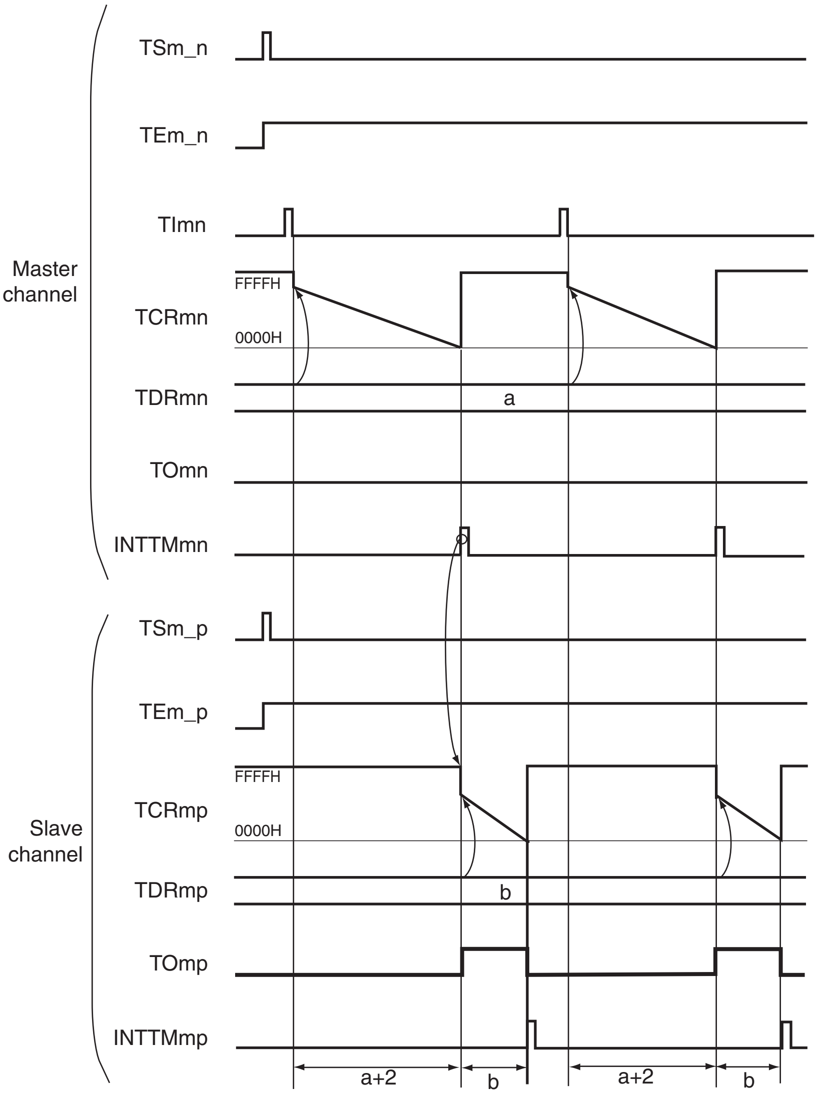
n = 0, 2, 4, 6

When m = 2:

n = 0, 2: 78K0R/HE3, 78K0R/HF3

n = 0, 2, 4, 6: 78K0R/HG3

<R> **Figure 6-70. Example of Basic Timing of Operation as One-Shot Pulse Output Function**



Remark m: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)

When m = 0: n = 0, 2, 4, 6

When m = 1:

n = 0, 2, 4, 6

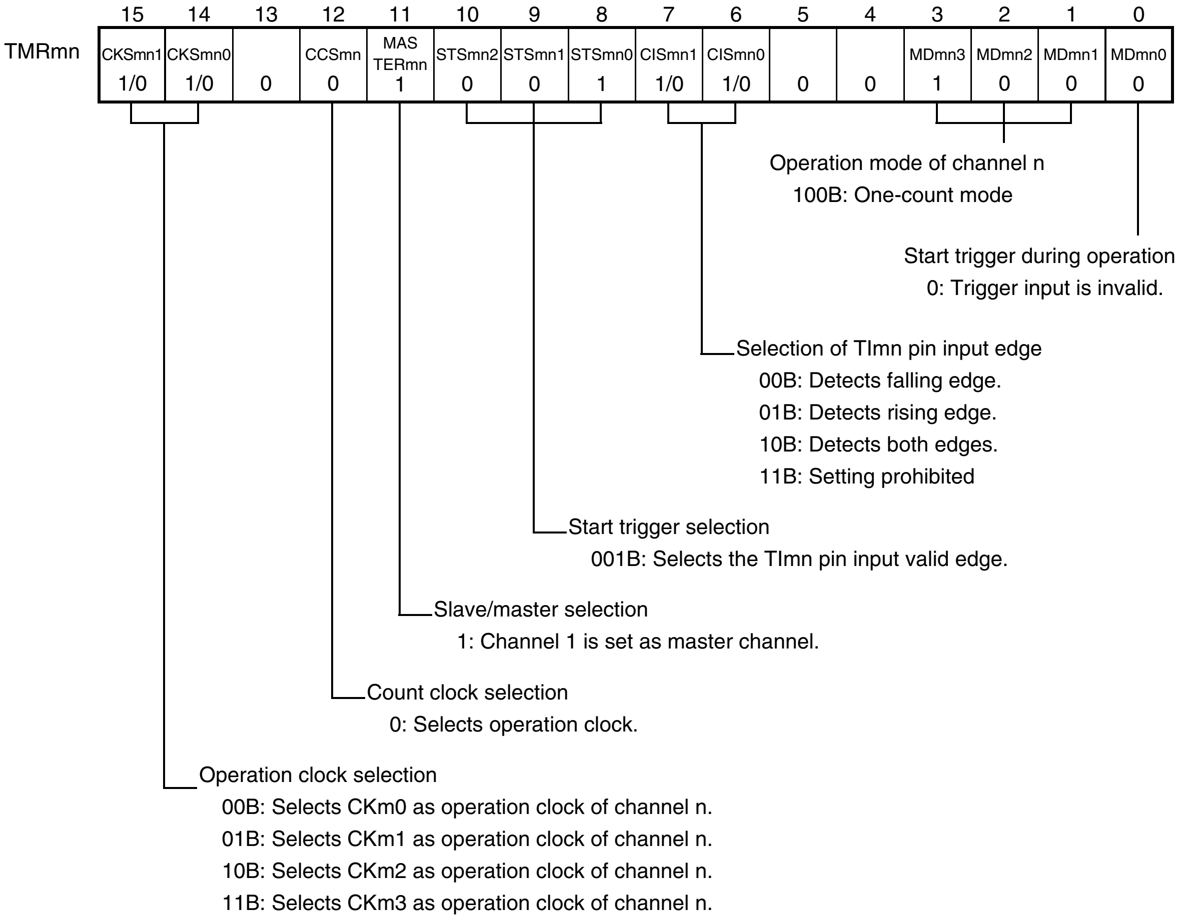
When m = 2:

n = 0, 2: 78K0R/HE3, 78K0R/HF3

n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-71. Example of Set Contents of Registers
When One-Shot Pulse Output Function Is Used (Master Channel) (1/2)

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	1	0	0	1	1/0	1/0	0	0	1	0	0	0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TImn pin input valid edge.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
00B: Selects CKm0 as operation clock of channel n.
01B: Selects CKm1 as operation clock of channel n.
10B: Selects CKm2 as operation clock of channel n.
11B: Selects CKm3 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEm_n
	0

0: Stops the TOmn output operation by counting operation.

Remark m: Unit number, n: Master channel number
When m = 0: n = 0, 2, 4, 6
When m = 1: n = 0, 2, 4, 6
When m = 2: n = 0, 2: 78K0R/HE3, 78K0R/HF3, n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-71. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel) (2/2)

(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: Unit number, n: Master channel number
When m = 0: n = 0, 2, 4, 6
When m = 1: n = 0, 2, 4, 6
When m = 2: n = 0, 2: 78K0R/HE3, 78K0R/HF3, n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-72. Example of Set Contents of Registers
When One-Shot Pulse Output Function Is Used (Slave Channel) (1/2)

(a) Timer mode register mp (TMRmp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	MAS TERmp	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	1/0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

Operation mode of channel p
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
00B: Selects CKm0 as operation clock of channel p.
01B: Selects CKm1 as operation clock of channel p.
10B: Selects CKm2 as operation clock of channel p.
11B: Selects CKm3 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p
TOm	TOmp
	1/0

0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p
TOEm	TOEm_p
	1/0

0: Stops the TOmp output operation by counting operation.
1: Enables the TOmp output operation by counting operation.

Remark m: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)
When m = 0: n = 0, 2, 4, 6
When m = 1: n = 0, 2, 4, 6
When m = 2: n = 0, 2: 78K0R/HE3, 78K0R/HF3, n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-72. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel) (2/2)

(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: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)
When m = 0: n = 0, 2, 4, 6
When m = 1: n = 0, 2, 4, 6
When m = 2: n = 0, 2: 78K0R/HE3, 78K0R/HF3, n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-73. 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.)
	The TAUmEN bit of PER0 register is set as 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 to CKm3.	
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 TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEm_p 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.

Remark m: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)

When m = 0: n = 0, 2, 4, 6

When m = 1:

n = 0, 2, 4, 6

When m = 2:

n = 0, 2: 78K0R/HE3, 78K0R/HF3

n = 0, 2, 4, 6: 78K0R/HG3

Figure 6-73. Operation Procedure of One-Shot Pulse Output Function (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets TOEm_p (slave) to 1 (only when operation is resumed). The TSm_n (master) and TSm_p (slave) bits of the TSm register are set to 1 at the same time. The TSm_n and TSm_p bits automatically return to 0 because they are trigger bits.	TEm_n and TEm_p are set to 1 and the master channel enters the TImn input edge detection wait status. Counter stops operating.
	Detects the TImn pin input valid edge of master channel.	Master channel starts counting.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. <R> Set values of the TMRmp registers, TOMmn, TOMmp, TOLmn, TOmn, and TOEmn bits cannot be changed. <R> TDRmn register is rewritable immediately after INTTMmn generating. <R> TDRmp register is rewritable immediately after INTTMmp generating. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. <R> Set values of the TOLmp, TOmp, and TOEmp bits can be changed.	Master channel loads the value of TDRmn to TCRmn when the TImn pin valid input edge 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 TTm_n (master) and TTm_p (slave) bits are set to 1 at the same time. The TTm_n and TTm_p bits automatically return to 0 because they are trigger bits.	TEm_n, TEm_p = 0, and count operation stops. TCRmn and TCRmp hold count value and stops. The TOmp output is not initialized but holds current status.
	TOEm_p of slave channel is cleared to 0 and value is set to the TOmp bit.	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears TOmp bit to 0 after the value to be held is set to the port register. When holding the TOmp pin output level is not necessary Switches the port mode register to input mode.	The TOmp pin output level is held by port function. The TOmp pin output level goes into Hi-Z output state.
	The TAUmEN bit of PER0 register is set as 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m: Unit number, n: Master channel number, p: Slave channel number (where p are integers greater than n+1)

When m = 0: n = 0, 2, 4, 6

When m = 1:

n = 0, 2, 4, 6

When m = 2:

n = 0, 2: 78K0R/HE3, 78K0R/HF3

n = 0, 2, 4, 6: 78K0R/HG3

6.8.3 Operation as multiple PWM output function

By extending the PWM function and using two or more slave channels, many PWM output signals can be produced.

For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDRmp (slave 1)}/{Set value of TDRmn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRmq (slave 2)}/{Set value of TDRmn (master) + 1} × 100

Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

TCRmn of the master channel operates in the interval timer mode and counts the periods.

TCRmp of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. TCRmp loads the value of TDRmp to TCRmp, using INTTMmn of the master channel as a start trigger, and start counting down. When TCRmp = 0000H, TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

In the same way as TCRmp of the slave channel 1, TCRmq of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. TCRmq loads the value of TDRmq to TCRmq, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = 0000H, TCRmq outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.

When channel 0 is used as the master channel as above, up to seven types of PWM signals can be generated for the timer array units 0 to 2.

Caution To rewrite both TDRmn of the master channel and TDRmp of the slave channel 1, write access is necessary at least twice. Since the values of TDRmn and TDRmp are loaded to TCRmn and TCRmp after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both TDRmn of the master and TDRmp of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to TDRmq of the slave channel 2).

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

Remark m: Unit number, n: Channel number, p: Slave channel number 1 (where p are consecutive integers greater than n + 1), q: Slave channel number 2 (where q are consecutive integers greater than n + 2)

When m = 0

$n = 0, 2, 4$

$n < p < q \leq 7$

When m = 1

$n = 0, 2, 4$

$n < p < q \leq 7$

When m = 2

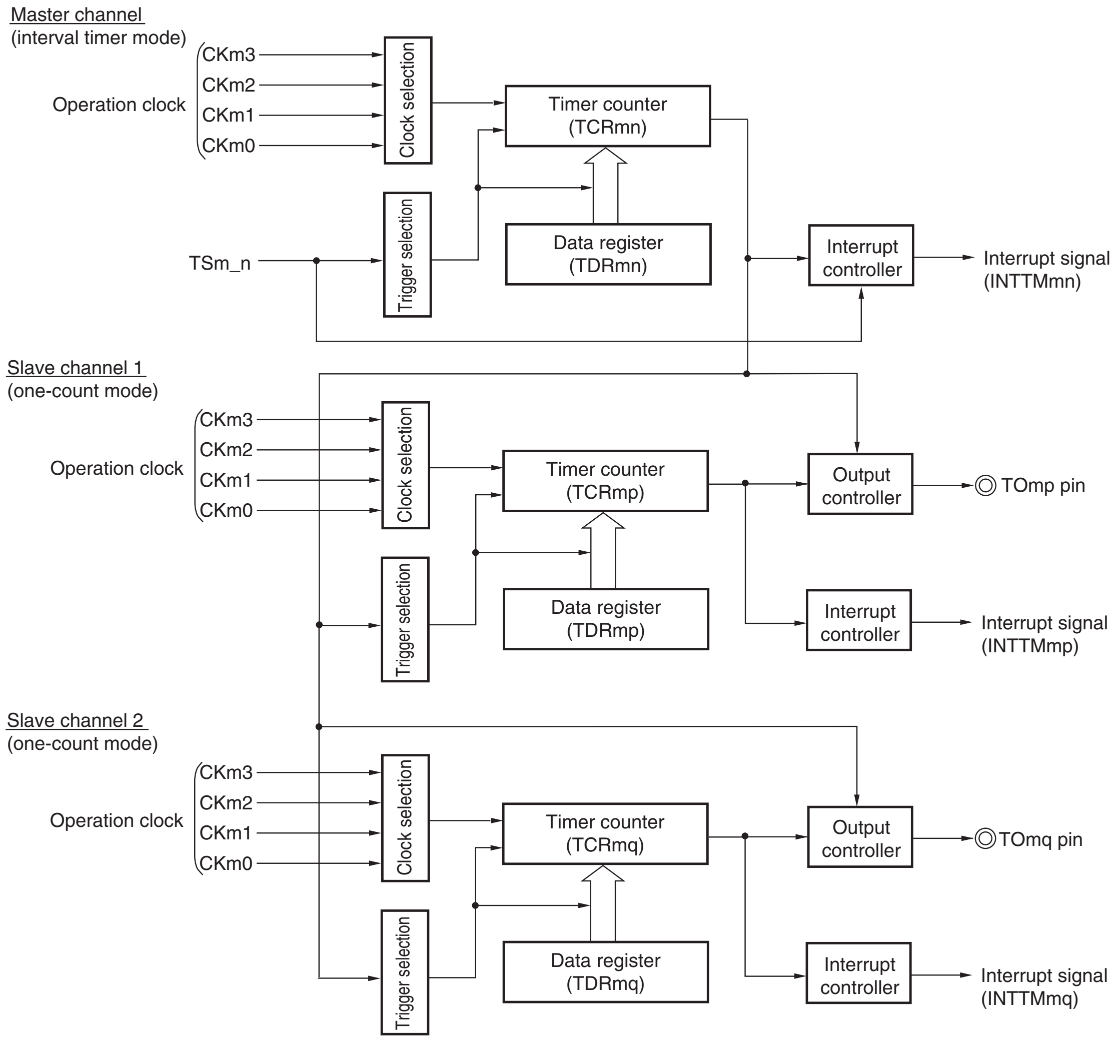
n = 0: 78K0R/HE3, 78K0R/HF3

$n < p < q \leq 3$

n = 0, 2, 4: 78K0R/HG3

$n < p < q \leq 7$

<R> **Figure 6-74. Block Diagram of Operation as Multiple PWM Output Function (Output Two Types of PWMs)**



Remark m: Unit number, n: Channel number, p: Slave channel number 1 (where p are consecutive integers greater than n + 1), q: Slave channel number 2 (where q are consecutive integers greater than n + 2)

When m = 0

n = 0, 2, 4

n < p < q ≤ 7

When m = 1

n = 0, 2, 4

n < p < q ≤ 7

When m = 2

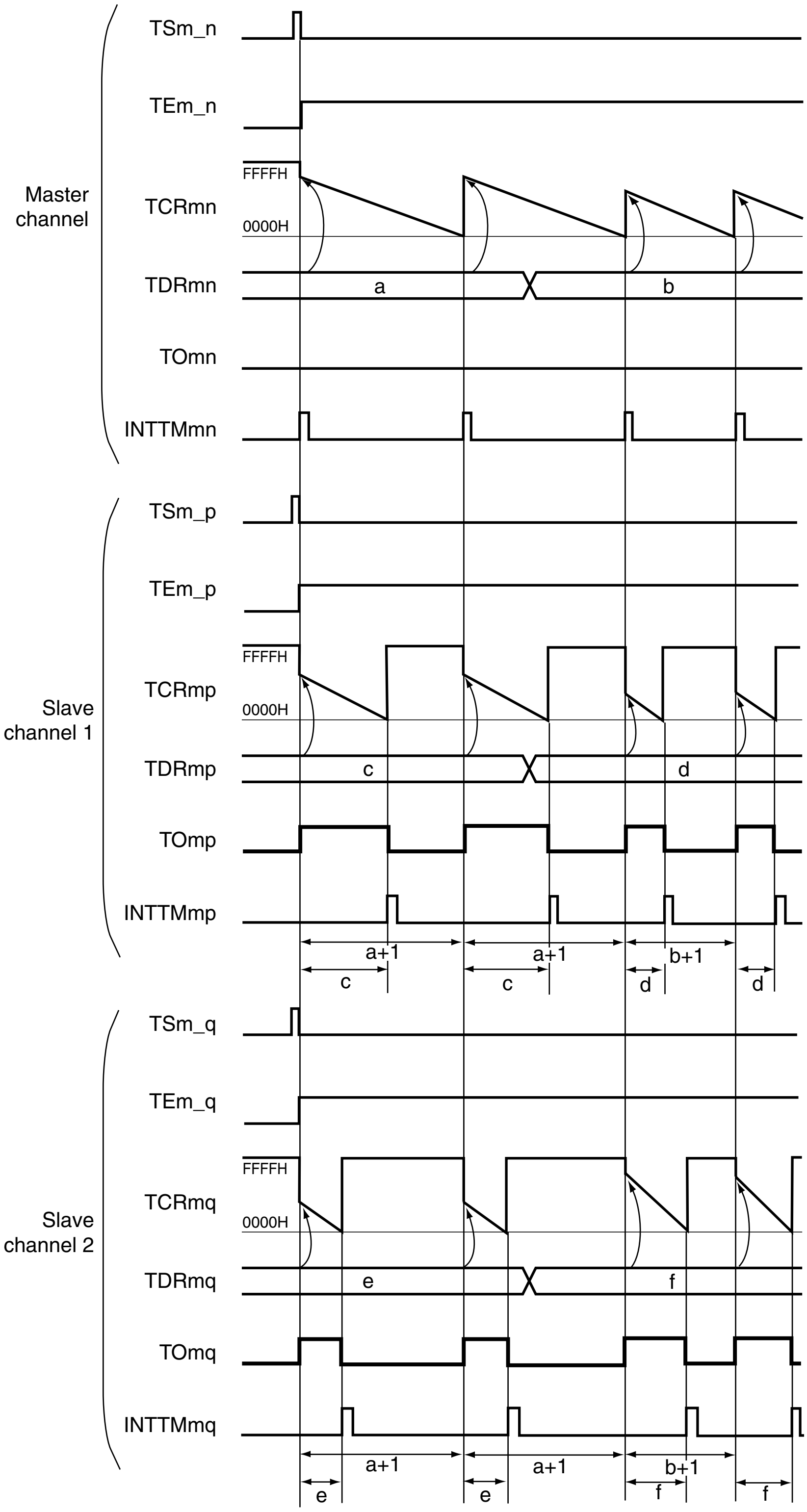
n = 0: 78K0R/HE3, 78K0R/HF3

n < p < q ≤ 3

n = 0, 2, 4: 78K0R/HG3

n < p < q ≤ 7

Figure 6-75. Example of Basic Timing of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



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

Remark m: Unit number, n: Channel number, p: Slave channel number 1 (where p are consecutive integers greater than n + 1), q: Slave channel number 2 (where q are consecutive integers greater than n + 2)

When m = 0

n = 0, 2, 4

$n < p < q \leq 7$

When m = 1

n = 0, 2, 4

$n < p < q \leq 7$

When m = 2

n = 0: 78K0R/HE3, 78K0R/HF3

$n < p < q \leq 3$

n = 0, 2, 4: 78K0R/HG3

$n < p < q \leq 7$

Figure 6-76. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used (1/2)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
00B: Selects CKm0 as operation clock of channel n.
01B: Selects CKm1 as operation clock of channel n.
10B: Selects CKm2 as operation clock of channel n.
11B: Selects CKm3 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEm_n
	0

0: Stops the TOmn output operation by counting operation.

Remark m: Unit number, n: Channel number
When m = 0:
n = 0, 2, 4
When m = 1:
n = 0, 2, 4
When m = 2:
n = 0: 78K0R/HE3, 78K0R/HF3
n = 0, 2, 4: 78K0R/HG3

Figure 6-76. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used (2/2)

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (toggle mode).

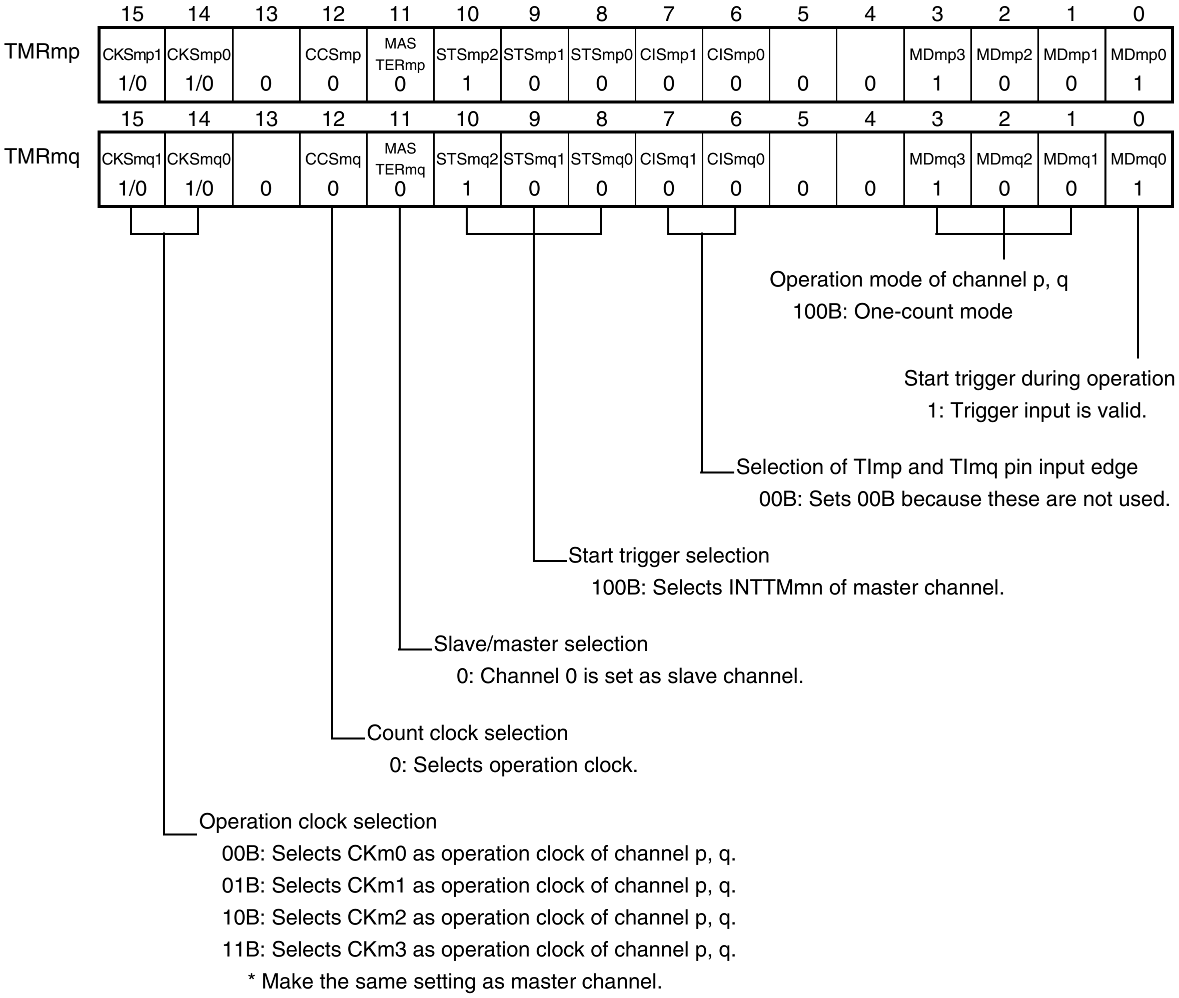
(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets toggle mode.

Remark m: Unit number, n: Channel number
When m = 0:
n = 0, 2, 4
When m = 1:
n = 0, 2, 4
When m = 2:
n = 0: 78K0R/HE3, 78K0R/HF3
n = 0, 2, 4: 78K0R/HG3

Figure 6-77. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (Output Two Types of PWMs) (1/2)

(a) Timer mode register mp, mq (TMRmp, TMRmq)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	MAS TERmp	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	1/0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmq	CKSmq1	CKSmq0		CCSmq	MAS TERmq	STSmq2	STSmq1	STSmq0	CISmq1	CISmq0			MDmq3	MDmq2	MDmq1	MDmq0
	1/0	1/0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p, q
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp and TImq pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
00B: Selects CKm0 as operation clock of channel p, q.
01B: Selects CKm1 as operation clock of channel p, q.
10B: Selects CKm2 as operation clock of channel p, q.
11B: Selects CKm3 as operation clock of channel p, q.
* Make the same setting as master channel.

Remark m: Unit number, n: Channel number, p: Slave channel number 1 (where p are consecutive integers greater than n + 1), q: Slave channel number 2 (where q are consecutive integers greater than n + 2)

When m = 0
n = 0, 2, 4
$n < p < q \leq 7$
When m = 1
n = 0, 2, 4
$n < p < q \leq 7$
When m = 2
n = 0: 78K0R/HE3, 78K0R/HF3
$n < p < q \leq 3$
n = 0, 2, 4: 78K0R/HG3
$n < p < q \leq 7$

Figure 6-77. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (Output Two Types of PWMs) (2/2)

(b) Timer output register m (TOm)

	Bit q	Bit p
TOm	TOmq 1/0	TOmp 1/0

0: Outputs 0 from TOmp or TOmq.
1: Outputs 1 from TOmp or TOmq.

(c) Timer output enable register m (TOEm)

	Bit q	Bit p
TOEm	TOEm_q 1/0	TOEm_p 1/0

0: Stops the TOmp or TOmq output operation by counting operation.
1: Enables the TOmp or TOmq output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit q	Bit p
TOLm	TOLmq 1/0	TOLmp 1/0

0: Positive logic output (active-high)
1: Inverted output (active-low)

(e) Timer output mode register m (TOMm)

	Bit q	Bit p
TOMm	TOMmq 1	TOMmp 1

1: Sets the combination operation mode.

Remark m: Unit number, n: Channel number, p: Slave channel number 1 (where p are consecutive integers greater than n + 1), q: Slave channel number 2 (where q are consecutive integers greater than n + 2)

When m = 0
n = 0, 2, 4
n < p < q ≤ 7
When m = 1
n = 0, 2, 4
n < p < q ≤ 7
When m = 2
n = 0: 78K0R/HE3, 78K0R/HF3
n < p < q ≤ 3
n = 0, 2, 4: 78K0R/HG3
n < p < q ≤ 7

Figure 6-78. Operation Procedure When Multiple PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	The TAUmEN bit of PER0 register is set as 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 to CKm3.	
Channel default setting	Sets the TMRmn, TMRmp, and TMRmq registers of each channel 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 and TDRmq registers of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp and TOMmq bits of the TOMm register are set to 1 (combination operation mode). TOLmp and a TOLmq bit are set up. Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. →	The TOmp and TOmq pins go into Hi-Z output state. The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets TOEm_p and TOEm_q to 1 and enables operation of TOmp and TOmq. →	TOmp or TOmq does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.
Operation start	Sets TOEm_p and TOEm_q (slave) to 1 (only when operation is resumed). The TSm_n bit (master), and TSm_p and TSm_q (slave) bits of the TSm register are set to 1 at the same time. → The TSm_n, TSm_p, and TSm_q bits automatically return to 0 because they are trigger bits.	TEm_n = 1, TEm_p, TEm_q = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.

Operation is resumed (on the next page).

Figure 6-78. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

	Software Operation	Hardware Status
During <R> operation	Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOmn, and TOEmn bits cannot be changed. Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated. The TCRmn, TCRmp, and TCRmq registers can always be read. The TSRmn, TSRmp, and TSRmq registers are not used. <R> Set values of the TOLmp, TOLmq, TOmp, TOmq, TOEmp, and TOEmq bits can 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 1, the values of TDRmp are transferred to TCRmp, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. At the slave channel 2, the values of TDRmq are transferred to TCRmq, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTm_n bit (master), TTm_p, and TTm_q (slave) bits are set to 1 at the same time. The TTm_n, TTm_p, and TTm_q bits automatically return to 0 because they are trigger bits.	TEm_n, TEm_p, and TEm_q = 0, and count operation stops. TCRmn, TCRmp, and TCRmq hold count value and stop. The TOmp and TOmq outputs are not initialized but hold current status.
	TOEm_p or TOEm_q of slave channel is cleared to 0 and value is set to the TOmp and TOmq bits.	The TOmp and TOmq pins output the TOmp and TOmq set levels.
TAU stop	To hold the TOmp and TOmq pin output levels Clears TOmp and TOmq bits to 0 after the value to be held is set to the port register. When holding the TOmp and TOmq pin output levels is not necessary Switches the port mode register to input mode.	The TOmp and TOmq pin output levels are held by port function. The TOmp and TOmq pin output levels go into Hi-Z output state.
	The TAUmEN bit of PER0 register is set as 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.)

Operation is resumed (on the before page).

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

Remark m: Unit number, n: Channel number, p: Slave channel number 1 (where p are consecutive integers greater than n + 1), q: Slave channel number 2 (where q are consecutive integers greater than n + 2)

When m = 0

$n = 0, 2, 4$

$n < p < q \leq 7$

When m = 1

$n = 0, 2, 4$

$n < p < q \leq 7$

When m = 2

$n = 0$: 78K0R/HE3, 78K0R/HF3

$n < p < q \leq 3$

$n = 0, 2, 4$: 78K0R/HG3

$n < p < q \leq 7$

CHAPTER 7 16-BIT WAKEUP TIMER

The 78K0R/Hx3 incorporates a 16-bit wakeup timer (WUTM).

7.1 Overview

The 16-bit wakeup timer (WUTM) has the following functions.

- Interval function
  Counter × 1
  Compare × 1
  Compare match interrupt × 1

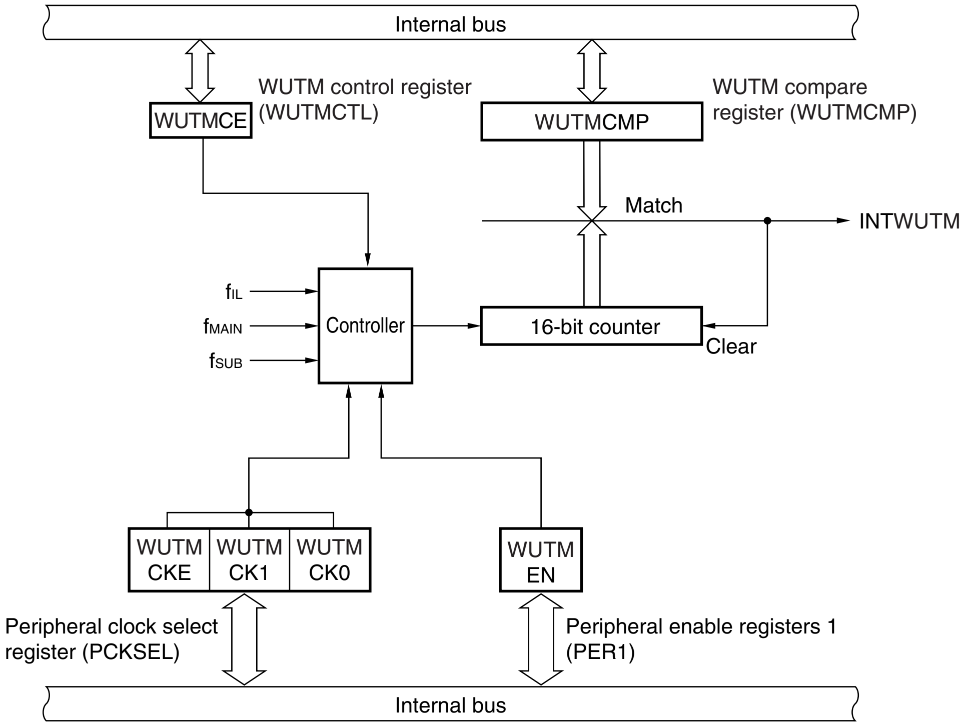
7.2 Configuration

WUTM includes the following hardware.

Table 7-1. Configuration of WUTM

Item	Configuration
Timer register	16-bit counter
Control register	Peripheral enable register 1 (PER1) Peripheral clock select register (PCKSEL) WUTM control register (WUTMCTL)
Register	WUTM compare register (WUTMCMP)

Figure 7-1. Block Diagram of WUTM


Internal bus
WUTM control register (WUTMCTL)
WUTMCE
WUTMCMP
WUTM compare register (WUTMCMP)
Match
INTWUTM
f_{IL}
f_{MAIN}
f_{SUB}
Controller
16-bit counter
Clear
WUTM CKE
WUTM CK1
WUTM CK0
WUTM EN
Peripheral clock select register (PCKSEL)
Peripheral enable registers 1 (PER1)
Internal bus

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

f_{MAIN}: Main system clock frequency

f_{SUB}: Subclock frequency

7.3 Register

(1) Peripheral enable register 1 (PER1)

These registers are 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.

PER1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears theses registers to 00H.

Caution Whether to enable or disable SFR writing only is selected for the 16-bit wakeup timer. Whether to enable or disable supplying the operating clock is selected using the PCKSEL register.

Figure 7-2. Format of Peripheral Enable Register 1 (PER1)

Address: F00F1H After reset: 00H R/W

Symbol	7	6	5	4	<3>	2	<1>	<0>
PER1	0	0	0	0	SAU2EN	0	WUTEN	DFLEN

<R>

WUTEN	Control of 16-bit wakeup timer input clock
0	Stops input clock supply for SFR writing. • SFR used by the 16-bit wakeup timer cannot be written.
1	Supplies input clock for SFR writing. • SFR used by the 16-bit wakeup timer can be read and written.

Caution Be sure to clear the following bits to 0.
78K0R/HC3: bits 2 to 7
78K0R/HE, 78K0R/HF3, 78K0R/HG3: bits 2, 4 to 7

(2) Peripheral clock select register (PCKSEL)

This register is used to select for and supply to each peripheral hardware device the operating clock.

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

Reset signal generation clears this register to 00H.

Caution Set the PCKSEL register before starting to operate each peripheral hardware device.

Figure 7-3. Format of Peripheral Clock Select Register (PCKSEL)

Address: F00F2H After reset: 00H R/W

Symbol	7	6	5	<4>	3	<2>	1	0
PCKSEL	0	0	0	CANMCKE	0	WUTMCKE	WUTMCK1	WUTMCK0

WUTMCKE	Control of 16-bit wakeup timer operating clock
0	Stops supplying operating clock.
1	Supplies operating clock.

WUTMCK1	WUTMCK0	16-bit wakeup timer operating clock selection
0	0	f_{IL}
0	1	f_{SUB}
1	0	$f_{MAIN}/2^8$
1	1	$f_{MAIN}/2^{12}$

Cautions 1. Be sure to clear bits 3, 5 to 7 of the PCKSEL register to 0.

<R> **2. Please change WUTMMCK1-0 after stopping a 16-bit wake up timer (WUTMCKE = 0).**

(3) WUTM compare register (WUTMCMP)

The WUTMCMP register is a 16-bit compare register.

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

Reset sets this register to 0000H.

Cautions 1. Rewriting the WUTMCMP register is prohibited while the timer is operating (WUTMCE = 1).

2. When writing the WUTMCMP register, be sure to set bit 1 (WUTEN) of peripheral enable register 1 (PER1) to 1 and supply an input clock.

Figure 7-4. WUTM compare register (WUTMCMP)

Address: FFFAEH, FFFAFH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUTMCMP																

(4) WUTM control register (WUTMCTL)

The WUTMCTL register is an 8-bit register that controls the WUTM operation.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

Caution When writing the WUTMCTL register, be sure to set bit 1 (WUTEN) of peripheral enable register 1 (PER1) to 1 and supply an input clock.

Figure 7-5. Format of WUTM control register (WUTMCTL)

Address: FFFACH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
WUTMCTL	WUTMCE	0	0	0	0	0	0	0

<table>
<tr><th>WUTMCE</th><th>Control of WUTM operation</th></tr>
<tr><td>0</td><td>Operation disabled</td></tr>
<tr><td>1</td><td>Operation enabled</td></tr>
<tr><td colspan="2">WUTM is asynchronously reset by the WUTMCE bit.
If the WUTMCE bit is set to “1”, the internal operating clock is enabled within two input clocks after the WUTMCE bit has been set to “1” and WUTM starts counting up.</td></tr>
</table>

Cautions 1. Be sure to clear bits 0 to 6 of the WUTMCTL register to 0.

2. If the WUTMCE bit is cleared to 0, the counter value within WUTM is immediately cleared.

7.4 Operation

7.4.1 Interval timer mode

In the interval timer mode, if the 16-bit counter and WUTM compare register (WUTMCMP) values match, the counter is <R> cleared to 0000H and starts counting up again at the same time a match interrupt signal (INTWUTM) is output.

Figure 7-6. Interval Timer Mode Operation Start Flow

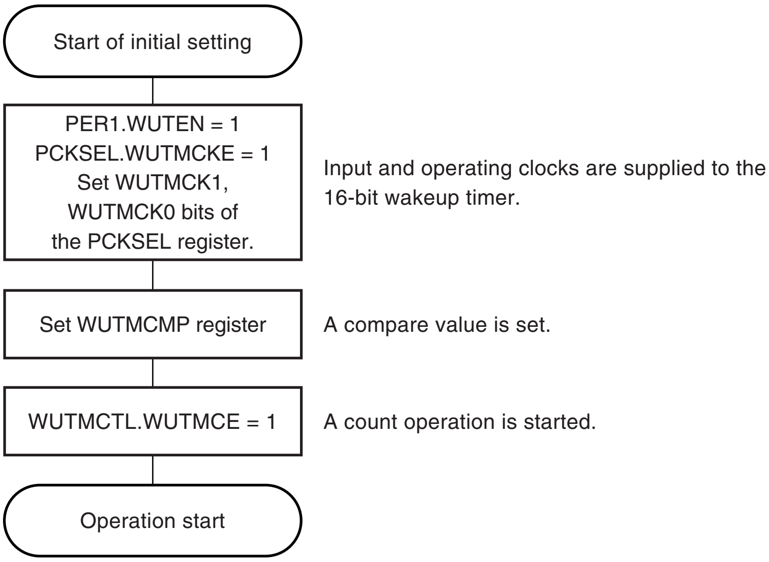
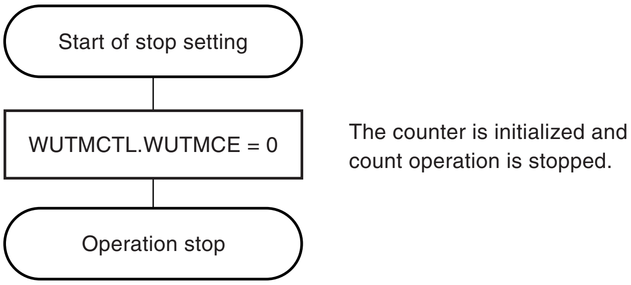
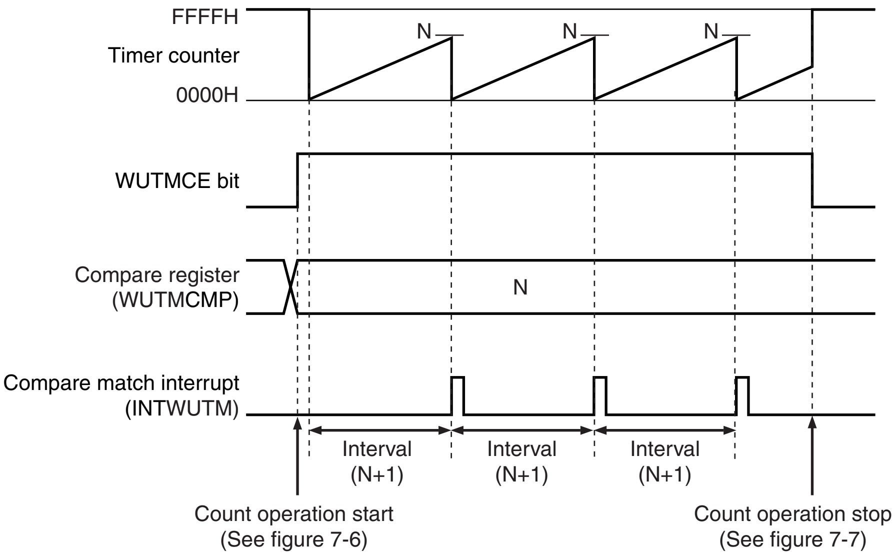


Figure 7-7. Interval Timer Mode Operation Stop Flow



Remark **To reduce the power consumption by stopping WUTM, clear (0) also PER1.WUTEN and PCKSEL.WUTMCKE.**

Figure 7-8. Operation Timing of Interval Timer Mode



Caution The interrupt cycle can be calculated by using the following expression.

INTWUTM (timer interrupt) generation cycle = Operating clock cycle × (WUTMCMP setting value + 1)

7.4.2 Cautions

(1) Clock generator and clock enable timing

The operation timing of the count clock is shown below.

Figure 7-9. Count Operation Start Timing (Min. Delay)

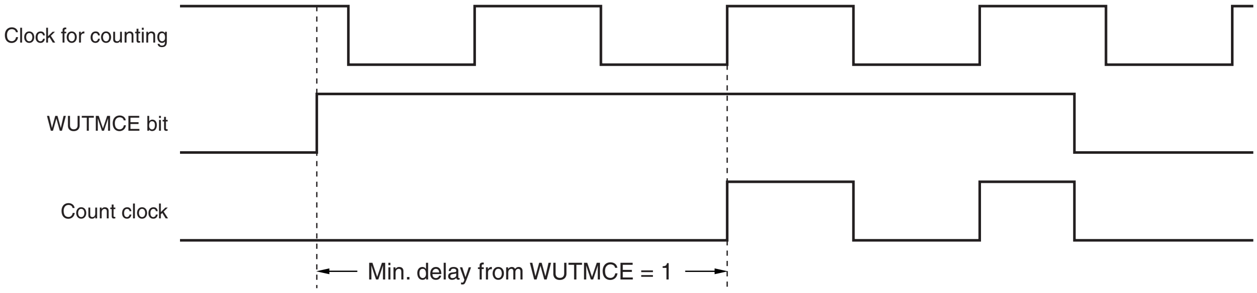
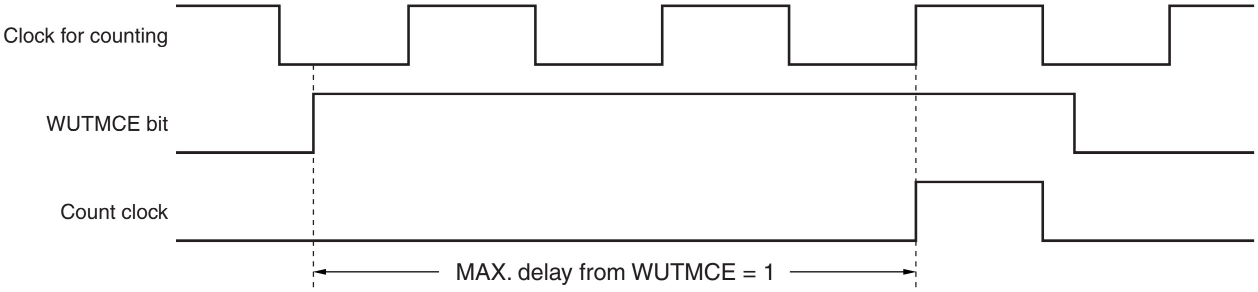


Figure 7-10. Count Operation Start Timing (Max. Delay)



(2) Rewriting register during WUTM operation

Rewriting the WUTMCMP register is prohibited while WUTM is operating.

If the WUTMCMP register is rewritten when the WUTMCE bit is 1, an interrupt may be generated.

CHAPTER 8 WATCHDOG TIMER

8.1 Functions of Watchdog Timer

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 19 RESET FUNCTION**.

When 75% of the overflow time is reached, an interval interrupt can be generated.

8.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 8-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

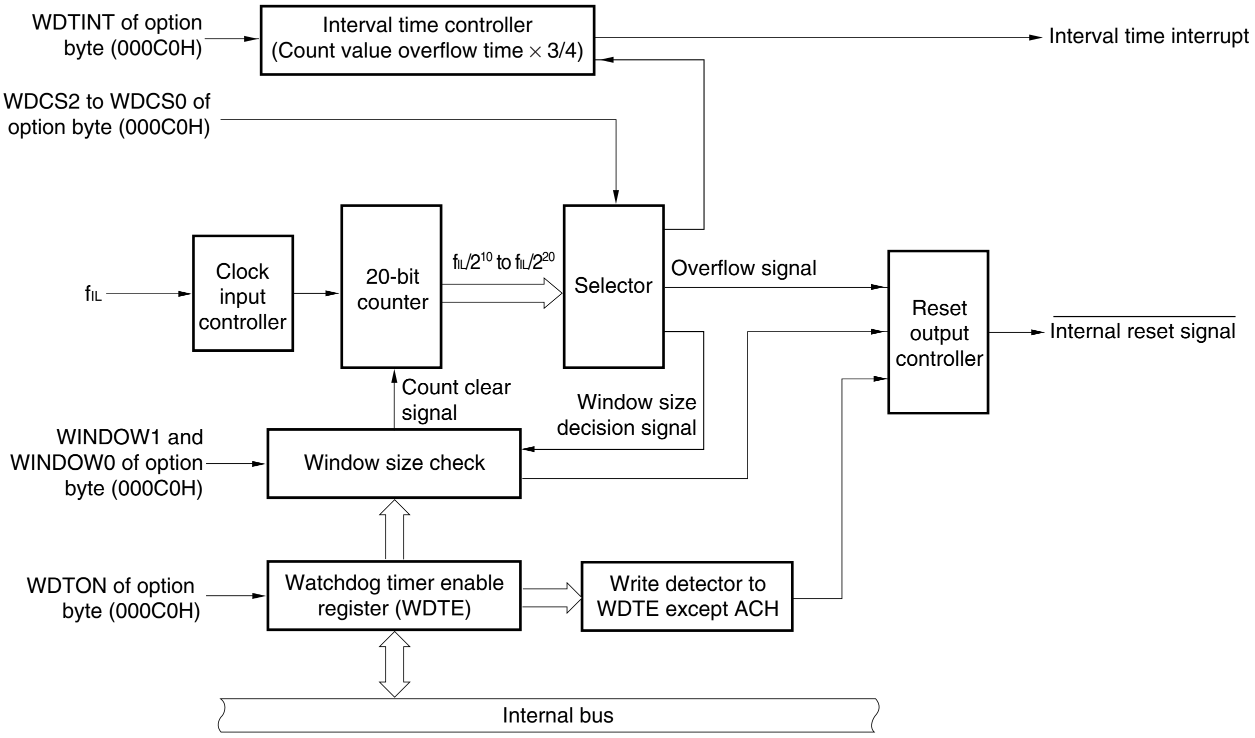
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 8-2. 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 23 OPTION BYTE**.

Figure 8-1. Block Diagram of Watchdog Timer


WDTINT of option byte (000C0H)
Interval time controller (Count value overflow time × 3/4)
Interval time interrupt
WDCS2 to WDCS0 of option byte (000C0H)
fIL
Clock input controller
20-bit counter
fIL/2^10 to fIL/2^20
Selector
Overflow signal
Reset output controller
Internal reset signal
Count clear signal
Window size decision signal
WINDOW1 and WINDOW0 of option byte (000C0H)
Window size check
WDTON of option byte (000C0H)
Watchdog timer enable register (WDTE)
Write detector to WDTE except ACH
Internal bus

Remark fIL: Internal low-speed oscillation clock frequency

8.3 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 8-2. 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)).

(2) Register used to change the operating mode of the watchdog timer during self programming (WDTSELF)

Writing “35H” to WDTSELF changes the overflow time to the maximum value and prevents the watchdog timer from being reset during self programming.

Caution When writing to the WDTSELF register, set the GDWDT bit of the GUARD register to “1”.

Figure 8-3. Format of Register used to change the operating mode of the watchdog timer during self programming (WDTSELF)

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WDTSELF								

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.

WDTSELF Setting Value	Watchdog timer overflow time during self programming
35H	$2^{17}/f_{IL}$
Other than 35H	Value set by bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H)

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

8.4 Operation of Watchdog Timer

8.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H, 000C1H).

   - 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 23**).

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 **8.4.2** and **CHAPTER 23**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **8.4.3** and **CHAPTER 23**).
   - Enable use of the internal low-speed oscillator by using bit 7 (LIOUSE) of the option byte (000C1H) (for details, see **CHAPTER 5** and **CHAPTER 23**).

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.
   A 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. The watchdog timer cannot be used as long as the internal low-speed oscillator is not set to an operating state by using the option byte.

2. If the internal low-speed oscillator is stopped during a STOP state by using the option byte, the watchdog timer is stopped during a STOP state regardless of the setting of WDSTBYON. The watchdog timer, however, is reset when it is stopped when WDSTBYON = 0, but the internal low-speed oscillator is stopped only and the watchdog timer retains the counter value before the STOP mode was set when WDSTBYON = 1. Start a count operation again after releasing the STOP mode.

3. 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.

4. 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.

5. 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 3FFH.

Cautions 6. 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.

7. 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 The following table shows the watchdog timer operating states according to the combinations of the option byte and CPU states.

Option Byte (000C1H)		Option Byte (000C0H)		CPU Status	Internal Low-Speed Oscillation Clock Operating State	Watchdog Timer Operating State
LIOUSE	LIOSTOPB	WDTON	WDSTBYON			
0	×	×	×	×	Stops	Stops
1	0	0	×	RUN/HALT	Operating	Stops
				STOP	Stops	Stops
1	0	1	0	RUN	Operating	Operating
				HALT	Operating	Stops
				STOP	Stops	Stops
1	0	1	1	RUN/HALT	Operating	Operating
				STOP	Stops	Stops
1	1	0	×	×	Operating	Stops
1	1	1	0	RUN	Operating	Operating
				HALT	Operating	Stops
				STOP	Operating	Stops
1	1	1	1	×	Operating	Operating

Remark ×: don't care

8.4.2 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 8-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (when f_{IL} = 30 kHz(TYP.))
0	0	0	$2^7/f_{IL}$ (4.27 ms)
0	0	1	$2^8/f_{IL}$ (8.53 ms)
0	1	0	$2^9/f_{IL}$ (17.07 ms)
0	1	1	$2^{10}/f_{IL}$ (34.13 ms)
1	0	0	$2^{12}/f_{IL}$ (136.5 ms)
1	0	1	$2^{14}/f_{IL}$ (546.1 ms)
1	1	0	$2^{15}/f_{IL}$ (1092 ms)
1	1	1	$2^{17}/f_{IL}$ (4369 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.

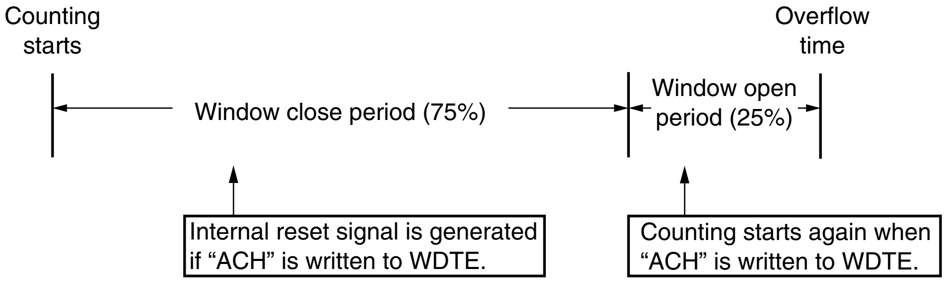
Remark f_{IL}: Internal low-speed oscillation clock frequency

8.4.3 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 25%


Counting starts
Overflow time
Window close period (75%)
Window open period (25%)
Internal reset signal is generated if “ACH” is written to WDTE.
Counting starts again when “ACH” is written to WDTE.

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 8-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	25%
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.

3. Do not set the window open period to 25% if the watchdog timer corresponds to either of the conditions below.

- **When used at a supply voltage (V_{DD}) below 2.7 V.**
- **When stopping all main system clocks (internal high-speed oscillation clock, X1 clock, and external main system clock) by use of the STOP mode or software.**

Remark If the overflow time is set to $2^{10}/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period			
	25%	50%	75%	100%
Window close time	0 to 28.44 ms	0 to 18.96 ms	0 to 9.48 ms	None
Window open time	28.44 to 31.03 ms	18.96 to 31.03 ms	9.48 to 31.03 ms	0 to 31.03 ms

<When window open period is 25%>

- 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 − 0.25) = 0 to $2^{10}/27$ kHz (MIN.) × 0.75 = 0 to 28.44 ms
- Window open time:
  $2^{10}/f_{IL}$ (MIN.) × (1 − 0.25) to $2^{10}/f_{IL}$ (MAX.) = $2^{10}/27$ kHz (MIN.) × 0.75 to $2^{10}/33$ kHz (MAX.) = 28.44 to 31.03 ms

8.4.4 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 8-5. 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 9 CLOCK OUTPUT CONTROLLER

9.1 Functions of Clock Output Controller

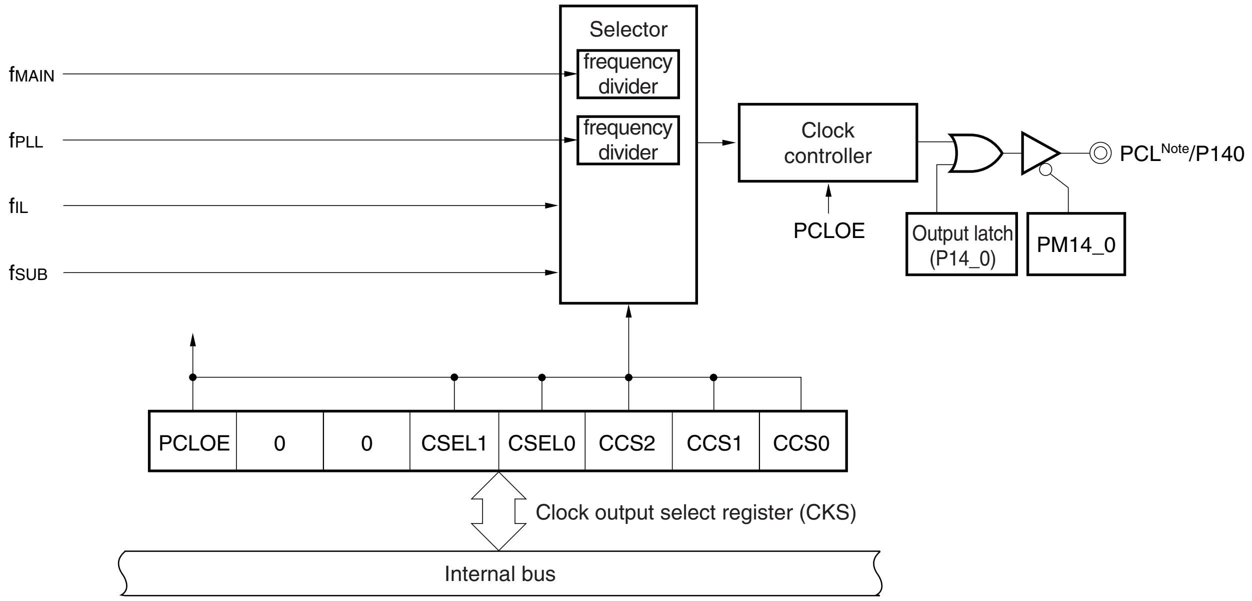
The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral ICs.

Clocks are output from the PCL pin.

PCL outputs a clock selected by clock output select register (CKS).

Figure 9-1 shows the block diagram of clock output controller.

Figure 9-1. Block Diagram of Clock Output Controller



Note The PCL pin can output a clock of up to 12 MHz.

Remark f_{MAIN}: Main system clock
f_{IL}: Internal low-speed oscillation clock
f_{PLL}: PLL clock
f_{SUB}: Subclock

9.2 Configuration of Clock Output Controller

The clock output controller includes the following hardware.

Table 9-1. Configuration of Clock Output Controller

Item	Configuration
Control registers	Clock output select registers (CKS) Port mode register 14 (PM14) Port register 14 (P14)

The initial value of the CKS after a reset is released can be set by using the option byte.

Table 9-2. Option Byte and Clock Output Controller Settings

Clock Output Controller Setting	Option Byte (000C2H)
Setting of enabling or disabling PCL output when a reset is released	Bit 7 (PCLOFF)
PCL output clock setting when a reset is released	Bits 4 to 0 (CSEL1B, CSEL0B, CCS2B to CCS0B)

Remark See **CHAPTER 23 OPTION BYTE** for details of the option byte.

9.3 Registers Controlling Clock Output Controller

The following two registers are used to control the clock output controller.

- Clock output select registers (CKS)
- Port mode register 14 (PM14)

(1) Clock output select register (CKS)

This register set output enable/disable for clock output pin (PCL), and set the output clock.
Select the clock to be output from PCL by using CKS.
CKS is set by a 1-bit or 8-bit memory manipulation instruction.

Figure 9-2. Format of Clock Output Select Register (CKS)

Address: FFFA5H After reset: xxH[Note 1] R/W

Symbol	<7>	6	5	4	3	2	1	0
CKS	PCLOE	0	0	CSEL1	CSEL0	CCS2	CCS1	CCS0

PCLOE	PCL output enable/disable specification
0	Output disable
1	Output enable

CSEL1	CSEL0	CCS2	CCS1	CCS0	PCL output clock selection				
						f_{MAIN} = 8 MHz	f_{PLL} = 16 MHz	f_{MAIN} = 20 MHz	f_{PLL} = 24 MHz
0	0	0	0	0	f_{MAIN}	8MHz	–	Setting prohibited[Note 2]	–
0	0	0	0	1	$f_{MAIN}/2$	4 MHz	–	10 MHz	–
0	0	0	1	0	$f_{MAIN}/2^2$	2 MHz	–	5 MHz	–
0	0	0	1	1	$f_{MAIN}/2^3$	1 MHz	–	2.5 MHz	–
0	0	1	0	0	$f_{MAIN}/2^4$	0.5 MHz	–	1.25 MHz	–
0	0	1	0	1	$f_{MAIN}/2^{11}$	3.91 kHz	–	9.76 kHz	–
0	0	1	1	0	$f_{MAIN}/2^{12}$	1.95 kHz	–	4.88 kHz	–
0	0	1	1	1	$f_{MAIN}/2^{13}$	0.98 kHz	–	2.44 kHz	–
0	1	0	0	0	f_{PLL}	–	Setting prohibited[Note 2]	–	Setting prohibited[Note 2]
0	1	0	0	1	$f_{PLL}/2$	–	8MHz	–	12 MHz
0	1	0	1	0	$f_{PLL}/2^2$	–	4 MHz	–	6 MHz
0	1	0	1	1	$f_{PLL}/2^3$	–	2 MHz	–	3 MHz
0	1	1	0	0	$f_{PLL}/2^4$	–	1 MHz	–	1.5 MHz
<R> 0	1	1	0	1	$f_{PLL}/2^{11}$	–	7.81 kHz	–	11.72 kHz
0	1	1	1	0	$f_{PLL}/2^{12}$	–	3.91 kHz	–	5.86 kHz
0	1	1	1	1	$f_{PLL}/2^{13}$	–	1.95 kHz	–	2.93 kHz
1	0	0	0	0	f_{IL}	30 kHz			
1	1	0	0	0	f_{SUB}	See **CHAPTER 29 ELECTRICAL SPECIFICATIONS**			
Other than above					Setting prohibited				

Notes 1. The initial value is the value set by using the option byte. See **CHAPTER 23 OPTION BYTE** for details.

2. Setting an output clock exceeding 12 MHz is prohibited

(**Cautions 1 to 3** and **Remarks 1 to 6** are given on the next page.)

Cautions 1. Change the output clock after disabling clock output (PCLOE = 0).

2. If the selected clock (f_{MAIN} or f_{PLL} or f_{IL} or f_{SUB}) stops during clock output (PCLOE = 1), the output becomes undefined.

3. Errors may be caused in the PCL output frequency due to fluctuation of EV_{DD} or EV_{SS}. Perform a thorough evaluation when a highly accurate clock is required.

Remarks 1. Outputting a clock can be started by setting the option byte after releasing a reset. See **CHAPTER 23 OPTION BYTE** for details.

2. The output clock and f_{CLK} are asynchronous. Use the output function of the timer array unit to output a clock that is synchronized with f_{CLK}.

3. f_{MAIN}: Main system clock frequency

4. f_{PLL}: PLL clock frequency

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

6. f_{SUB}: Subclock frequency

(2) Port mode register 14 (PM14)

This register sets port 14 input/output in 1-bit units.

When using the P140/PCL pin for clock output, clear PM14_0 and the output latch of P14_0 to 0.

PM14 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FEH.

<R> **Figure 9-3. Format of Port Mode Register 14 (PM14)**

Address: FFF2EH After reset: FEH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	1	1	1	1	1	PM14_0

PM14_0	P140 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

9.4 Operations of Clock Output Controller

Output pin, PCL, is available.
PCL outputs a clock selected by clock output select register (CKS).

9.4.1 Operation as output pin

PCL is output as the following procedure.

<1> Select the output frequency with bits 0 to 4 (CCS0 to CCS2, CSEL0, and CSEL1) of the clock output select register (CKS) of the PCL pin (output in disabled status).

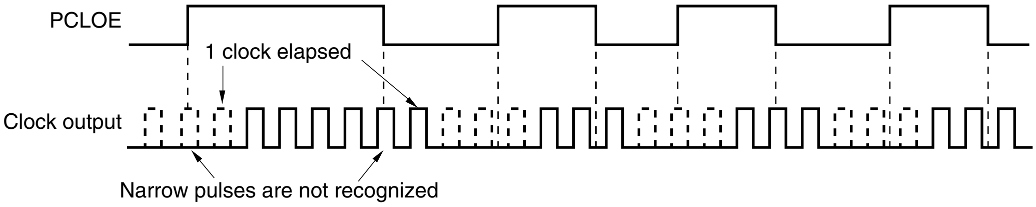
<2> Set bit 7 (PCLOE) of CKS to 1 to enable clock output.

Caution To shifting the STOP mode, executing STOP instruction at least 1/2 period of the output clock after stopping clock output. If a clock that stops oscillating in STOP mode is selected, a pulse with a narrow width may be output when transitioning to STOP mode.

Remarks 1. The clock output controller starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOE) is switched. At this time, pulses with a narrow width are not output. **Figure 9-4** shows enabling or stopping output using PCLOE and the timing of outputting the clock.

2. The initial value is the value set by using the option byte. See **CHAPTER 23 OPTION BYTE** for details.

Figure 9-4. Output Application Example



CHAPTER 10 A/D CONVERTER

	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
Analog input channels	11	15	16	24

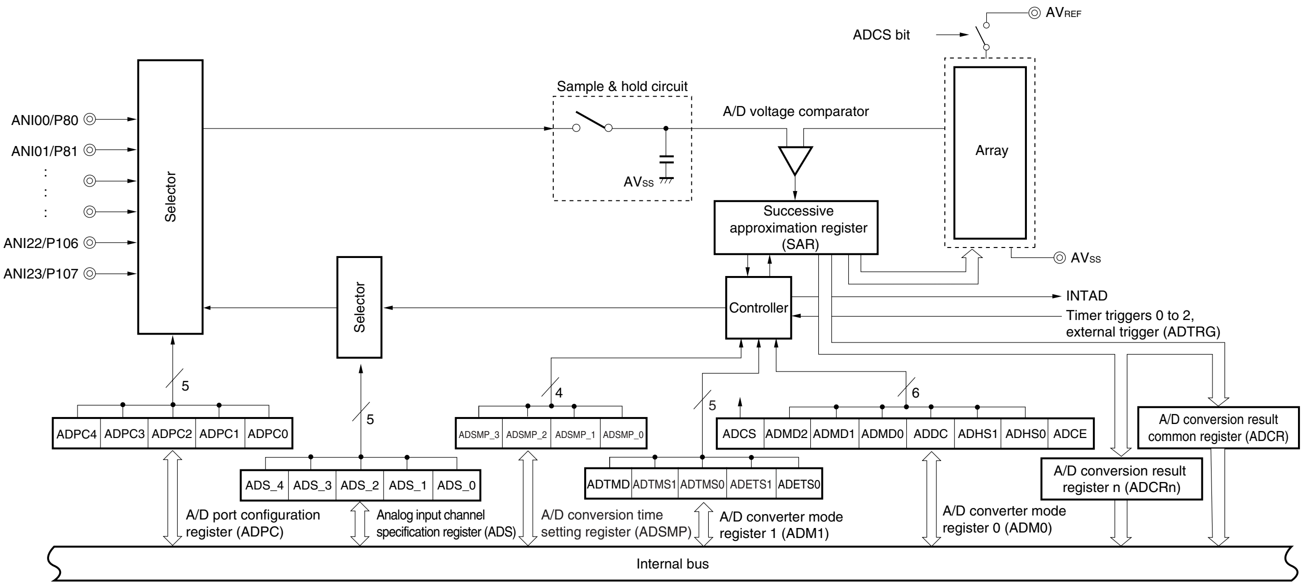
10.1 Function of A/D Converter

The A/D converter is a 10-bit resolution converter that converts analog input signals into digital values, and is configured to control up to 24 channels of A/D converter analog inputs (ANI00 to ANI23).

The A/D converter has the following function.

- **10-bit resolution A/D conversion**
  10-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI00 to ANI23. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Figure 10-1. Block Diagram of A/D Converter


AVREF
ADCS bit
Sample & hold circuit
A/D voltage comparator
ANI00/P80
ANI01/P81
ANI22/P106
ANI23/P107
Selector
AVSS
Array
Successive approximation register (SAR)
AVSS
Controller
Selector
INTAD
Timer triggers 0 to 2, external trigger (ADTRG)
5
5
4
5
6
ADPC4 ADPC3 ADPC2 ADPC1 ADPC0
ADSMP_3 ADSMP_2 ADSMP_1 ADSMP_0
ADCS ADMD2 ADMD1 ADMD0 ADDC ADHS1 ADHS0 ADCE
A/D conversion result common register (ADCR)
ADS_4 ADS_3 ADS_2 ADS_1 ADS_0
ADTMD ADTMS1 ADTMS0 ADETS1 ADETS0
A/D conversion result register n (ADCRn)
A/D port configuration register (ADPC)
Analog input channel specification register (ADS)
A/D conversion time setting register (ADSMP)
A/D converter mode register 1 (ADM1)
A/D converter mode register 0 (ADM0)
Internal bus

Remark n = 0 to 10, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI10/P92: 78K0R/HC3
n = 0 to 14, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI14/P96: 78K0R/HE3
n = 0 to 15, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97: 78K0R/HF3
n = 0 to 23, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97, ANI16/P100 to ANI23/P107: 78K0R/HG3

10.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI00 to ANI23 pins

These are the analog input pins of the 24-channel 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.

Remark	ANI00 to ANI10:	78K0R/HC3
	ANI00 to ANI14:	78K0R/HE3
	ANI00 to ANI15:	78K0R/HF3
	ANI00 to ANI23:	78K0R/HG3

(2) ADTRG pin

This is the external trigger input pin of the A/D converter.

(3) 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.

(4) A/D voltage comparator

This A/D voltage comparator compares the voltage generated from the voltage tap of the array with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REF}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set (1). If the analog input voltage is less than the reference voltage (1/2 AV_{REF}), the MSB of the SAR is cleared (0).
After that, bit 10 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the array is selected by the value of bit 9 (MSB), to which the result has been already set.

Bit 9 = 0: (1/4 AV_{REF})
Bit 9 = 1: (3/4 AV_{REF})

The voltage tap of the array and the analog input voltage are compared and bit 10 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of array: Bit 8 = 1
Analog input voltage ≤ Voltage tap of array: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.

(5) Array

The array generates the comparison voltage input from an analog input pin.

(6) Successive approximation register (SAR)

The SAR register is a 10-bit register that sets voltage tap data whose values from the array match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result common register (ADCR) and A/D conversion result register n (ADCRn). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(7) 10-bit A/D conversion result common register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(8) 8-bit A/D conversion result common 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 higher 8 bits of the A/D conversion result are stored.

(9) 10-bit A/D conversion result register n (ADCRn)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(10) 8-bit A/D conversion result register n (ADCRnH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the higher 8 bits of the A/D conversion result are stored.

(11) 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.

(12) AV_{REF} pin

This pin inputs the reference voltage of the A/D converter, the power supply pins and A/D converter of the comparator, and the comparator.

The analog signal input to ANI00 to ANI23 is converted into a digital signal, based on the voltage applied across AV_{REF} and AV_{SS}.

(13) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always use this pin at the same potential as that of the V_{SS} pin even when the A/D converter is not used.

Remark n = 0 to 10, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI10/P92: 78K0R/HC3
n = 0 to 14, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI14/P96: 78K0R/HE3
n = 0 to 15, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97: 78K0R/HF3
n = 0 to 23, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97, ANI16/P100 to ANI23/P107: 78K0R/HG3

10.3 Registers Controlling A/D Converter

The A/D converter is controlled by the following eleven registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- 10-bit A/D conversion result common register (ADCR)
- 8-bit A/D conversion result common register (ADCRH)
- 10-bit A/D conversion result register n (ADCRn)
- 8-bit A/D conversion result register n (ADCRnH)
- A/D port configuration register (ADPC)
- Analog input channel specification register (ADS)
- A/D conversion time setting register (ADSMP)
- Port mode registers 8 to 10 (PM8 to PM10)

Remark n = 0 to 10, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI10/P92: 78K0R/HC3
n = 0 to 14, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI14/P96: 78K0R/HE3
n = 0 to 15, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97: 78K0R/HF3
n = 0 to 23, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97, ANI16/P100 to ANI23/P107: 78K0R/HG3

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

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

Reset signal generation clears this register to 00H.

Figure 10-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	ADCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN

ADCEN	Control of A/D converter input clock
0	Stops supply of input clock. • SFR used by the A/D converter cannot be written (can be read). • The A/D converter is in the reset status[Note].
1	Supplies input clock. • SFR used by the A/D converter can be read/written.

Note The ADPC register is not reset even if PER0.ADCEN = 0 is set.

Cautions 1. When setting the A/D converter, be sure to set ADCEN to 1 first. If ADCEN = 0, writing to a control register of the A/D converter is ignored, and, even if the register is read, only the default value is read (except for port mode registers 8 to 10 (PM8 to PM10)).

2. In the 78K0R/HC3, be sure to clear bit 2 to "0".

(2) A/D converter mode register 0 (ADM0)

This register sets the specification of operation mode, 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 10-3. Format of A/D Converter Mode Register 0 (ADM0)

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM0	ADCS	ADMD2	ADMD1	ADMD0	ADDC	ADHS1	ADHS0	ADCE

ADCS	A/D conversion control
0	Stops A/D conversion
1	Enables A/D conversion

ADMD2	ADMD1	ADMD0	Specification of A/D converter operation mode
0	0	X	Continuous select mode
0	1	0	Continuous scan mode (interrupt generated for each channel)
0	1	1	Continuous scan mode (interrupt generated after conversion of channel, specified by ADS register)
1	0	X	One-shot select mode
1	1	0	One-shot scan mode (interrupt generated for each channel)
1	1	1	One-shot scan mode (interrupt generated after conversion of channel, specified by ADS register)

ADDC	Discharge function control[Note 1]
0	Discharge function disabled
1	Discharge function enabled

ADHS1	ADHS0	Specification of A/D conversion mode[Note 1]
0	0	High-speed mode 1 (4.0 V ≤ AV_{REF} ≤ 5.5 V)
0	1	High-speed mode 2 (2.7 V ≤ AV_{REF} ≤ 5.5 V)
1	0	Normal mode (2.7 V ≤ AV_{REF} ≤ 5.5 V)
1	1	Setting prohibited

ADCE	A/D voltage comparator operation control[Note 2]
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

Notes 1. For details of discharge function, and A/D conversion, see **(10) A/D conversion time setting register (ADSMP)**.

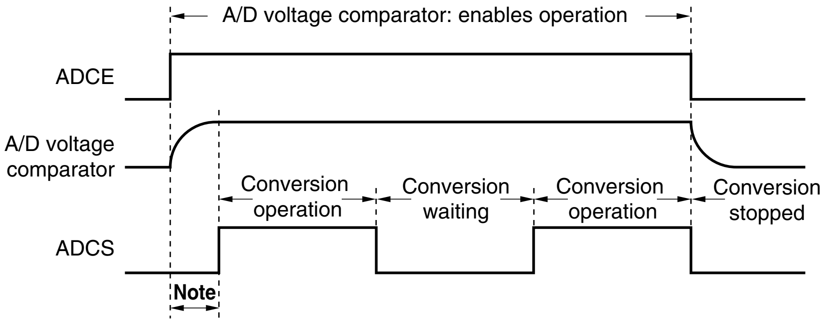
2. The operation of the A/D voltage comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

(**Cautions 1 to 3**, and **Remark** are given on the next page.)

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 (only A/D voltage comparator consumes power)
1	0	Setting prohibited
1	1	Conversion mode(A/D voltage comparator: enables operation)

Figure 10-4. 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 rising of the ADCS bit must be 1 μs or longer.

Cautions 1. The discharge function samples the V_{SS} voltage for a fixed period after A/D conversion ends (Only sampling is performed, comparison is not performed). Consequently, the A/D conversion time when the discharge function is enabled is longer than when the discharge function is disabled.

2. A/D conversion must be stopped before rewriting bits ADMD2 to ADMD0, ADDC, ADHS1, and ADHS0 to values other than the identical data.

3. The operations when the ADCS bit is set or cleared are as follows. Furthermore, to enable the first conversion result, set ADCE to 1 before starting an operation and set a conversion wait state.

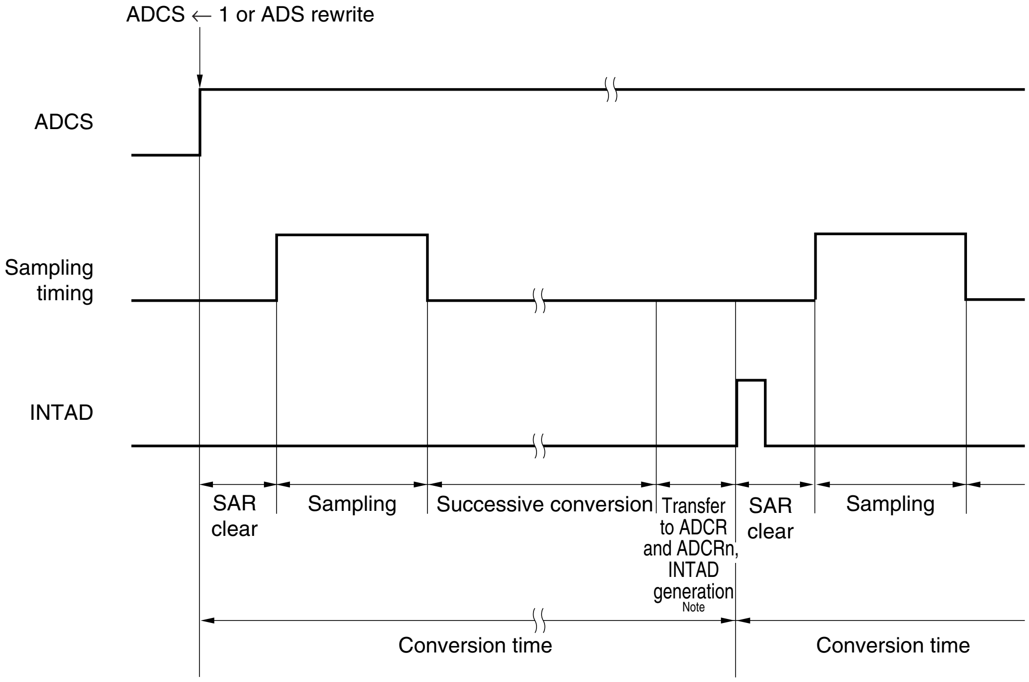
0 → 1: Starts a conversion operation.

1 → 1: Aborts conversion and starts re-conversion if the ADCS bit is overwritten by 1 during the conversion operation.
Restarts the conversion operation also after the completion of a one-shot operation.
Waits for a trigger in hardware trigger mode.

1 → 0: Stops the conversion operation if the ADCS bit is cleared to 0 during the conversion operation.

Remark n = 0 to 10: 78K0R/HC3
n = 0 to 14: 78K0R/HE3
n = 0 to 15: 78K0R/HF3
n = 0 to 23: 78K0R/HG3

Figure 10-5. A/D Converter Sampling and A/D Conversion Timing



Note When ADMD bit is 1

Remark n = 0 to 10: 78K0R/HC3

n = 0 to 14: 78K0R/HE3

n = 0 to 15: 78K0R/HF3

n = 0 to 23: 78K0R/HG3

(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 10-6. Format of A/D Converter Mode Register 1 (ADM1)

Address: FFF42H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
ADM1	ADTMD	0	0	0	ADTMS1	ADTMS0	ADETS1	ADETS0

ADTMD	A/D trigger mode selection
0	Software trigger mode (immediately starts conversion when ADCS = 1)
1	Hardware trigger mode (waits for trigger when ADCS = 1)

ADTMS1	ADTMS0	Timer trigger signal selection
0	0	Timer trigger signal 0 (INTTM12)
0	1	Timer trigger signal 1 (INTTM22)
1	0	Setting prohibited
1	1	External pin trigger signal (ADTRG)

ADETS1	ADETS0	Specification of external trigger (ADTRG pin) input valid edge
0	0	No edge detection (Trigger is not generated)
0	1	Falling edge detection
1	0	Rising edge detection
1	1	Detection of both rising and falling edges

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 common register (ADCR)

This register is a 16-bit register that stores the A/D conversion result in the select mode. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register. The 10 bits of the conversion result are stored in the higher 10 bits of the ADCR register, and 0 is read from the lower 6 bits.

ADCR can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 10-7. Format of 10-bit A/D Conversion Result Common Register (ADCR)

Address: FFF1EH, FFF1FH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR											0	0	0	0	0	0

Caution When writing to A/D converter mode register0, 1 (ADM0, ADM1), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR may become undefined. Read the conversion result before writing to ADM0, ADM1, ADS, and ADPC after the conversion operation of the target channel ends. Using timing other than the above may cause an incorrect conversion result to be read.

(5) 8-bit A/D conversion result common 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 in ADCRH.

ADCRH can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-8. Format of 8-bit A/D Conversion Result Common Register (ADCRH, ADCRL)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to A/D converter mode register0, 1 (ADM0, ADM1), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR may become undefined. Read the conversion result before writing to ADM0, ADM1, ADS, and ADPC after the conversion operation of the target channel ends. Using timing other than the above may cause an incorrect conversion result to be read.

(6) 10-bit A/D conversion result register n (ADCRn)

This register is a 16-bit register that stores the A/D conversion result in the select mode. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register. The 10-bit conversion result is stored in the higher 10 bits and 0 is read to the lower 6 bits of the ADCRn register.

ADCRn can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 10-9. Format of 10-bit A/D Conversion Result Register n (ADCRn)

Address: ADCR0 0280H, 0281H, ADCR1 0282H, 0283H, After reset: 0000H R
ADCR2 0284H, 0285H, ADCR3 0286H, 0287H,
ADCR4 0288H, 0289H, ADCR5 028AH, 028BH,
ADCR6 028CH, 028DH, ADCR7 028EH, 028FH,
ADCR8 0290H, 0291H, ADCR9 0292H, 0293H,
ADCR10 0294H, 0295H, ADCR11 0296H, 0297H,
ADCR12 0298H, 0299H, ADCR13 029AH, 029BH,
ADCR14 029CH, 029DH, ADCR15 029EH, 029FH,
ADCR16 02A0H, 02A1H, ADCR17 02A2H, 02A3H,
ADCR18 02A4H, 02A5H, ADCR19 02A6H, 02A7H,
ADCR20 02A8H, 02A9H, ADCR21 02AAH, 02ABH,
ADCR22 02ACH, 02ADH, ADCR23 02AEH, 02AFH

Symbol	0281H (In case of ADCR0)								0280H (In case of ADCR0)							
ADCRn											0	0	0	0	0	0

Caution When writing to A/D converter mode register0, 1 (ADM0, ADM1), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR may become undefined. Read the conversion result before writing to ADM0, ADM1, ADS, and ADPC after the conversion operation of the target channel ends. Using timing other than the above may cause an incorrect conversion result to be read.

Remark n = 0 to 10: 78K0R/HC3
n = 0 to 14: 78K0R/HE3
n = 0 to 15: 78K0R/HF3
n = 0 to 23: 78K0R/HG3

(7) 8-bit A/D conversion result register n (ADCRnH)

This register is an 8-bit register that stores the A/D conversion result.

The higher 8 bits of 10-bit resolution are stored in ADCRnH.

ADCRnH can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-10. Format of 8-bit A/D Conversion Result Register n (ADCRnH)

Address: ADCR0H 00281H, ADCR1H 00283H, After reset: 00H R
ADCR2H 00285H, ADCR3H 00287H,
ADCR4H 00289H, ADCR5H 0028BH,
ADCR6H 0028DH, ADCR7H 0028FH,
ADCR8H 00291H, ADCR9H 00283H,
ADCR10H 00295H, ADCR11H 00297H,
ADCR12H 00299H, ADCR13H 0029BH,
ADCR14H 0029DH, ADCR15H 0029FH,
ADCR16H 002A1H, ADCR17H 002A3H,
ADCR18H 002A5H, ADCR19H 002A7H,
ADCR20H 002A9H, ADCR21H 002ABH,
ADCR22H 002ADH, ADCR23H 002AFH

Symbol	7	6	5	4	3	2	1	0
ADCRnH								

Caution When writing to A/D converter mode register0, 1 (ADM0, ADM1), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR may become undefined. Read the conversion result before writing to ADM0, ADM1, ADS, and ADPC after the conversion operation of the target channel ends. Using timing other than the above may cause an incorrect conversion result to be read.

Remark n = 0 to 10: 78K0R/HC3
n = 0 to 14: 78K0R/HE3
n = 0 to 15: 78K0R/HF3
n = 0 to 23: 78K0R/HG3

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

This register switches the ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97 and ANI16/P100 to ANI23/P107 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 00H[Note].

Note The ADPC register is not reset even if PER0.ADCEN = 0 is set.

Figure 10-11. Format of A/D Port Configuration Register (ADPC)

Address: F0017H After reset: 00H[Note] R/W

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

ADPC bit					Analog input (A)/digital I/O (D) switching																							
4	3	2	1	0	ANI23/P107	ANI22/P106	ANI21/P105	ANI20/P104	ANI19/P103	ANI18/P102	ANI17/P101	ANI16/P100	ANI15/P97	ANI14/P96	ANI13/P95	ANI12/P94	ANI11/P93	ANI10/P92	ANI09/P91	ANI08/P90	ANI07/P87	ANI06/P86	ANI05/P85	ANI04/P84	ANI03/P83	ANI02/P82	ANI01/P81	ANI00/P80
0	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
0	0	0	0	1	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A
0	0	0	1	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A
0	0	0	1	1	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A
0	0	1	0	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A
0	0	1	0	1	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A
0	0	1	1	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A
0	0	1	1	1	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A
0	1	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A
0	1	0	0	1	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A
0	1	0	1	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A
0	1	0	1	1	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A
0	1	1	0	0	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A
0	1	1	0	1	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A
0	1	1	1	0	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A
0	1	1	1	1	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	0	0	0	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	0	0	1	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	0	1	0	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	0	1	1	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	1	0	0	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	1	0	1	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	1	1	0	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	0	1	1	1	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
1	1	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Other than above					Setting prohibited																							

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

Note The ADPC register is not reset even if PER0.ADCEN = 0 is set.

Cautions 1. Set a channel to be used for A/D conversion in the input mode by using port mode register 8 to 10 (PM8 to PM10).

2. Do not set the pin that is set by ADPC as digital I/O by ADS.

3. P80/ANI00 to P87/ANI07, P90/ANI08 to P97/ANI15 and P100/ANI16 to P107/ANI23 set as analog inputs in the order of P80/ANI00, ..., P87/ANI07, P90/ANI08, ..., P97/ANI15, P100/ANI16, ..., P107/ANI23 by the A/D port configuration register (ADPC). When using P80/ANI00 to P87/ANI07, P90/ANI08 to P97/ANI15 and P100/ANI16 to P107/ANI23 as analog inputs, start designing from P80/ANI00.

4. Be sure to first set the ADCEN bit of peripheral enable register 0 (PER0) to 1 when setting up the ADPC register. If ADCEN = 0, writing to the ADPC register is ignored and specified values are returned to the initial values.

Remark ANI00 to ANI10: 78K0R/HC3
ANI00 to ANI14: 78K0R/HE3
ANI00 to ANI15: 78K0R/HF3
ANI00 to ANI23: 78K0R/HG3

(9) 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.

<R> **Figure 10-12. 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	ADS_4	ADS_3	ADS_2	ADS_1	ADS_0

ADS_4	ADS_3	ADS_2	ADS_1	ADS_0	Select channel	
					Select mode	Scan mode
0	0	0	0	0	ANI00	ANI00
0	0	0	0	1	ANI01	ANI00-ANI01
0	0	0	1	0	ANI02	ANI00-ANI01-ANI02
0	0	0	1	1	ANI03	ANI00-ANI01…-ANI03
0	0	1	0	0	ANI04	ANI00-ANI01…-ANI04
0	0	1	0	1	ANI05	ANI00-ANI01…-ANI05
0	0	1	1	0	ANI06	ANI00-ANI01…-ANI06
0	0	1	1	1	ANI07	ANI00-ANI01…-ANI07
0	1	0	0	0	ANI08	ANI00-ANI01…-ANI08
0	1	0	0	1	ANI09	ANI00-ANI01…-ANI09
0	1	0	1	0	ANI10	ANI00-ANI01…-ANI10
0	1	0	1	1	ANI11	ANI00-ANI01…-ANI11
0	1	1	0	0	ANI12	ANI00-ANI01…-ANI12
0	1	1	0	1	ANI13	ANI00-ANI01…-ANI13
0	1	1	1	0	ANI14	ANI00-ANI01…-ANI14
0	1	1	1	1	ANI15	ANI00-ANI01…-ANI15
1	0	0	0	0	ANI16	ANI00-ANI01…-ANI16
1	0	0	0	1	ANI17	ANI00-ANI01…-ANI17
1	0	0	1	0	ANI18	ANI00-ANI01…-ANI18
1	0	0	1	1	ANI19	ANI00-ANI01…-ANI19
1	0	1	0	0	ANI20	ANI00-ANI01…-ANI20
1	0	1	0	1	ANI21	ANI00-ANI01…-ANI21
1	0	1	1	0	ANI22	ANI00-ANI01…-ANI22
1	0	1	1	1	ANI23	ANI00-ANI01…-ANI23
1	1	×	×	×	ANI00	ANI00

Cautions 1. Be sure to clear bits 5 to 7 to “0”.

2. Set a channel to be used for A/D conversion in the input mode by using port mode registers 8 to10 (PM8 to PM10).

3. Do not set the pin that is set by ADPC as digital I/O by ADS.

4. If a channel without an analog input is selected, the conversion result will be undefined.

5. Rewriting ADS during A/D conversion is prohibited. Rewrite ADS when the conversion operation has been stopped (ADCS = 0).

(**Remarks 1 and 2** are listed on the next page.)

Remarks 1. × : don't care

2. ANI00/P80 to ANI07/P87, ANI08/P90 to ANI10/P92: 78K0R/HC3
ANI00/P80 to ANI07/P87, ANI08/P90 to ANI14/P96: 78K0R/HE3
ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97: 78K0R/HF3
ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97, ANI16/P100 to ANI23/P107: 78K0R/HG3

(10) A/D conversion time setting register (ADSMP)

This register sets the A/D conversion time.

The setting value depends on the CPU/peripheral hardware clock frequency.

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

The number of conversion clocks of one channel is determined using ADSMP_3 to ADSMP_0. See **Table 10-2 Number of Conversion Clocks of One Channel (No Discharge)** and **Table 10-3 Number of Conversion Clocks of One Channel (Discharge Performed)** for details.

Reset signal generation clears this register to 00H.

<R> **Figure 10-13. Format of A/D Conversion Time Setting Register (ADSMP)**

Address: FFF33H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADSMP	0	0	0	0	ADSMP_3	ADSMP_2	ADSMP_1	ADSMP_0

Table 10-2. Number of Conversion Clocks per Channel (No Discharge)

ADSMP_3	ADSMP_2	ADSMP_1	ADSMP_0	CPU/Peripheral Hardware Clock Frequency	Number of Conversion Clocks per Channel		
					High-Speed Mode 1	High-Speed Mode 2	Normal Mode
0	0	0	0	$f_{CLK} \leq 2$ MHz	16 clocks	17 clocks	22 clocks
0	0	0	1	$f_{CLK} \leq 4$ MHz	18 clocks	20 clocks	40 clocks
0	0	1	0	$f_{CLK} \leq 6$ MHz	20 clocks	35 clocks	58 clocks
0	0	1	1	$f_{CLK} \leq 8$ MHz	33 clocks	37 clocks	76 clocks
0	1	0	0	$f_{CLK} \leq 10$ MHz	34 clocks	40 clocks	94 clocks
0	1	0	1	$f_{CLK} \leq 12$ MHz	37 clocks	56 clocks	114 clocks
0	1	1	0	$f_{CLK} \leq 14$ MHz	50 clocks	59 clocks	132 clocks
0	1	1	1	$f_{CLK} \leq 16$ MHz	52 clocks	72 clocks	150 clocks
1	0	0	0	$f_{CLK} \leq 18$ MHz	54 clocks	76 clocks	168 clocks
1	0	0	1	$f_{CLK} \leq 20$ MHz	55 clocks	78 clocks	186 clocks
1	0	1	0	$f_{CLK} \leq 22$ MHz	71 clocks	96 clocks	208 clocks
1	0	1	1	$f_{CLK} \leq 24$ MHz	73 clocks	99 clocks	226 clocks
1	1	0	0		75 clocks	113 clocks	244 clocks
1	1	0	1		88 clocks	116 clocks	262 clocks
1	1	1	0		89 clocks	118 clocks	280 clocks
1	1	1	1		92 clocks	133 clocks	300 clocks

Cautions 1. Be sure to clear bits 4 to 7 to "0".

2. Rewriting ADSMP during A/D conversion is prohibited. Rewrite ADS when the conversion operation has been stopped (ADCS = 0).

3. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

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

Table 10-3. Number of Conversion Clocks per Channel (Discharge Performed)

ADSMP_3	ADSMP_2	ADSMP_1	ADSMP_0	CPU/Peripheral Hardware Clock Frequency	Number of Conversion Clocks per Channel (Discharge Time Included)		
					High-Speed Mode 1	High-Speed Mode 2	Normal Mode
0	0	0	0	$f_{\text{CLK}} \leq 2$ MHz	19 clocks	22 clocks	27 clocks
0	0	0	1	$f_{\text{CLK}} \leq 4$ MHz	22 clocks	25 clocks	47 clocks
0	0	1	0	$f_{\text{CLK}} \leq 6$ MHz	24 clocks	42 clocks	68 clocks
0	0	1	1	$f_{\text{CLK}} \leq 8$ MHz	38 clocks	44 clocks	88 clocks
0	1	0	0	$f_{\text{CLK}} \leq 10$ MHz	39 clocks	48 clocks	108 clocks
0	1	0	1	$f_{\text{CLK}} \leq 12$ MHz	44 clocks	67 clocks	132 clocks
0	1	1	0	$f_{\text{CLK}} \leq 14$ MHz	58 clocks	70 clocks	152 clocks
0	1	1	1	$f_{\text{CLK}} \leq 16$ MHz	60 clocks	84 clocks	173 clocks
1	0	0	0	$f_{\text{CLK}} \leq 18$ MHz	63 clocks	89 clocks	193 clocks
1	0	0	1	$f_{\text{CLK}} \leq 20$ MHz	64 clocks	92 clocks	213 clocks
1	0	1	0	$f_{\text{CLK}} \leq 22$ MHz	83 clocks	115 clocks	240 clocks
1	0	1	1	$f_{\text{CLK}} \leq 24$ MHz	86 clocks	119 clocks	260 clocks
1	1	0	0		88 clocks	134 clocks	281 clocks
1	1	0	1		102 clocks	137 clocks	301 clocks
1	1	1	0		103 clocks	140 clocks	321 clocks
1	1	1	1		108 clocks	157 clocks	345 clocks

Cautions 1. Be sure to clear bits 4 to 7 to “0”.

2. Rewriting ADSMP during A/D conversion is prohibited. Rewrite ADSMP when the conversion operation has been stopped (ADCS = 0).

3. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark Tables 10-4 to 10-9 show the ADSMP register setting and the A/D conversion time in each mode.
The A/D conversion time of one channel is as follows.

Example: Continuous mode

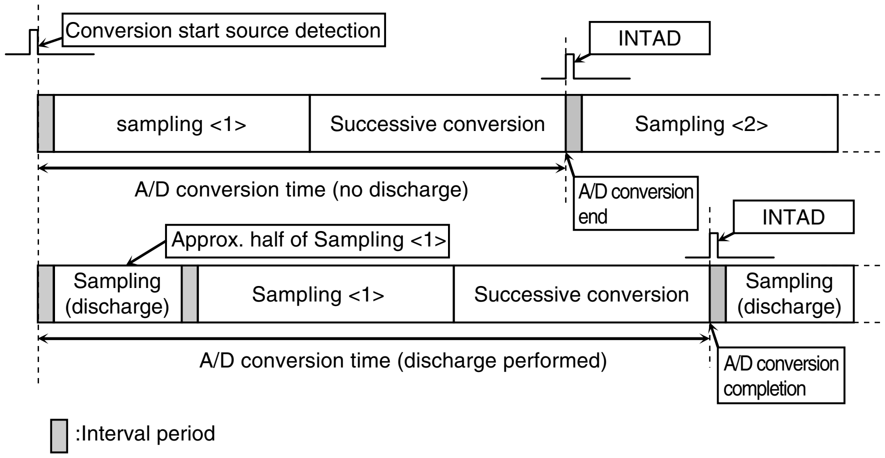


Table 10-4. A/D Conversion Time by CPU/Peripheral Hardware Clock Frequency (High-Speed Mode 1 : ADHS1, ADHS0 = 0, 0) (No Discharge)

ADSMP_3	ADSMP_2	ADSMP_1	ADSMP_0	A/D conversion time (μs) according to the CPU/peripheral hardware clock frequency				
				2 MHz	8 MHz	16 MHz	20 MHz	24 MHz
0	0	0	0	8 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	0	1	9 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	0	10 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	1	16.5 μs	4.125 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	0	17 μs	4.25 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	1	18.5 μs	4.625 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	0	25 μs	6.25 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	1	26 μs	6.5 μs	3.25 μs	Setting prohibited	Setting prohibited
1	0	0	0	27 μs	6.75μs	3.375 μs	Setting prohibited	Setting prohibited
1	0	0	1	27.5 μs	6.875 μs	3.437 μs	2.75 μs	Setting prohibited
1	0	1	0	35.5 μs	8.875 μs	4.438 μs	3.55 μs	Setting prohibited
1	0	1	1	36.5 μs	9.125 μs	4.563 μs	3.65 μs	3.042 μs
1	1	0	0	37.5 μs	9.375 μs	4.688 μs	3.75 μs	3.125 μs
1	1	0	1	44 μs	11 μs	5.5 μs	4.4 μs	3.667 μs
1	1	1	0	44.5 μs	11.125 μs	5.563 μs	4.45 μs	3.709 μs
1	1	1	1	46 μs	11.5 μs	5.75 μs	4.6 μs	3.834 μs

Table 10-5. A/D Conversion Time by CPU/Peripheral Hardware Clock Frequency (High-Speed Mode 1 : ADHS1, ADHS0 = 0, 0) (Discharge Performed)

ADSMP_3	ADSMP_2	ADSMP_1	ADSMP_0	A/D conversion time (μs) according to the CPU/peripheral hardware clock frequency				
				2 MHz	8 MHz	16 MHz	20 MHz	24 MHz
0	0	0	0	9.5 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	0	1	11 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	0	12 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	1	19 μs	4.75 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	0	19.5 μs	4.875 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	1	22 μs	5.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	0	29 μs	7.25 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	1	30 μs	7.5 μs	3.75 μs	Setting prohibited	Setting prohibited
1	0	0	0	31.5 μs	7.875 μs	3.937 μs	Setting prohibited	Setting prohibited
1	0	0	1	32 μs	8 μs	4 μs	3.2 μs	Setting prohibited
1	0	1	0	41.5 μs	10.375 μs	5.188 μs	4.15 μs	Setting prohibited
1	0	1	1	43 μs	10.75 μs	5.375 μs	4.3 μs	3.584 μs
1	1	0	0	44 μs	11 μs	5.5 μs	4.4 μs	3.667 μs
1	1	0	1	51 μs	12.75 μs	6.375 μs	5.1 μs	4.25 μs
1	1	1	0	51.5 μs	12.875 μs	6.438 μs	5.15 μs	4.292 μs
1	1	1	1	54 μs	13.5 μs	6.75 μs	5.4 μs	4.5 μs

Table 10-6. A/D Conversion Time by CPU/Peripheral Hardware Clock Frequency (High-Speed Mode 2 : ADHS1, ADHS0 = 0, 1) (No Discharge)

ADSMP_3	ADSMP_2	ADSMP_1	ADSMP_0	A/D conversion time (μs) according to the CPU/peripheral hardware clock frequency				
				2 MHz	8 MHz	16 MHz	20 MHz	24 MHz
0	0	0	0	8.5 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	0	1	10 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	0	17.5 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	1	18.5 μs	4.625 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	0	20 μs	5 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	1	28 μs	7 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	0	29.5 μs	7.375 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	1	36 μs	9 μs	4.5 μs	Setting prohibited	Setting prohibited
1	0	0	0	38.5 μs	9.5 μs	4.75 μs	Setting prohibited	Setting prohibited
1	0	0	1	39 μs	9.75 μs	4.875 μs	3.9 μs	Setting prohibited
1	0	1	0	48 μs	12 μs	6 μs	4.8 μs	Setting prohibited
1	0	1	1	49.5 μs	12.375 μs	6.188 μs	4.95 μs	4.125 μs
1	1	0	0	56.5 μs	14.125 μs	7.063 μs	5.65 μs	4.709 μs
1	1	0	1	58 μs	14.5 μs	7.25 μs	5.8 μs	4.834 μs
1	1	1	0	59 μs	14.75 μs	7.375 μs	5.9 μs	4.917 μs
1	1	1	1	66.5 μs	16.625 μs	8.313 μs	6.65 μs	5.542 μs

Table 10-7. A/D Conversion Time by CPU/Peripheral Hardware Clock Frequency (High-Speed Mode 2 : ADHS1, ADHS0 = 0, 1) (Discharge Performed)

ADSMP_3	ADSMP_2	ADSMP_1	ADSMP_0	A/D conversion time (μs) according to the CPU/peripheral hardware clock frequency				
				2 MHz	8 MHz	16 MHz	20 MHz	24 MHz
0	0	0	0	11 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	0	1	12.5 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	0	21 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	1	22 μs	5.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	0	24 μs	6 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	1	33.5 μs	8.375 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	0	35 μs	8.75 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	1	42 μs	10.5 μs	5.25 μs	Setting prohibited	Setting prohibited
1	0	0	0	45 μs	11.125 μs	5.562 μs	Setting prohibited	Setting prohibited
1	0	0	1	46 μs	11.5 μs	5.75 μs	4.6 μs	Setting prohibited
1	0	1	0	57.5 μs	14.375 μs	7.189 μs	5.75 μs	Setting prohibited
1	0	1	1	59.5 μs	14.875 μs	7.438 μs	5.95 μs	4.959 μs
1	1	0	0	67 μs	16.75 μs	8.375 μs	6.7 μs	5.584 μs
1	1	0	1	68.5 μs	17.125 μs	8.563 μs	6.85 μs	5.709 μs
1	1	1	0	70 μs	17.5 μs	8.75 μs	7 μs	5.834 μs
1	1	1	1	78.5 μs	19.625 μs	9.813 μs	7.85 μs	6.542 μs

Table 10-8. A/D Conversion Time by CPU/Peripheral Hardware Clock Frequency (Normal Mode : ADHS1, ADHS0 = 1, 0) (No Discharge)

ADSMP_3	ADSMP_2	ADSMP_1	ADSMP_0	A/D conversion time (μs) according to the CPU/peripheral hardware clock frequency				
				2 MHz	8 MHz	16 MHz	20 MHz	24 MHz
0	0	0	0	11 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	0	1	20 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	0	29 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	1	38 μs	9.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	0	47 μs	11.75 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	1	57 μs	14.25 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	0	66 μs	16.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	1	75 μs	18.75 μs	9.375 μs	Setting prohibited	Setting prohibited
1	0	0	0	Setting prohibited	21 μs	10.5 μs	Setting prohibited	Setting prohibited
1	0	0	1	Setting prohibited	23.25 μs	11.625 μs	9.3 μs	Setting prohibited
1	0	1	0	Setting prohibited	26 μs	13 μs	10.4 μs	Setting prohibited
1	0	1	1	Setting prohibited	28.25 μs	14.125 μs	11.3 μs	9.417 μs
1	1	0	0	Setting prohibited	30.5 μs	15.25 μs	12.2 μs	10.167 μs
1	1	0	1	Setting prohibited	32.75 μs	16.375 μs	13.1 μs	10.917 μs
1	1	1	0	Setting prohibited	35 μs	17.5 μs	14 μs	11.667 μs
1	1	1	1	Setting prohibited	37.5 μs	18.75 μs	15 μs	12.5 μs

Table 10-9. A/D Conversion Time by CPU/Peripheral Hardware Clock Frequency (Normal Mode : ADHS1, ADHS0 = 1, 0) (Discharge Performed)

ADSMP_3	ADSMP_2	ADSMP_1	ADSMP_0	A/D conversion time (μs) according to the CPU/peripheral hardware clock frequency				
				2 MHz	8 MHz	16 MHz	20 MHz	24 MHz
0	0	0	0	13.5 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	0	1	23.5 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	0	34 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	1	44 μs	11 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	0	54 μs	13.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	1	66 μs	16.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	0	76 μs	19 μs	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	1	86.5 μs	21.625 μs	10.813 μs	Setting prohibited	Setting prohibited
1	0	0	0	Setting prohibited	24.125 μs	12.063 μs	Setting prohibited	Setting prohibited
1	0	0	1	Setting prohibited	26.625 μs	13.313 μs	10.65 μs	Setting prohibited
1	0	1	0	Setting prohibited	30 μs	15 μs	12 μs	Setting prohibited
1	0	1	1	Setting prohibited	32.5 μs	16.25 μs	13 μs	10.834 μs
1	1	0	0	Setting prohibited	35.125 μs	17.563 μs	14.05 μs	11.709 μs
1	1	0	1	Setting prohibited	37.625 μs	18.813 μs	15.05 μs	12.542 μs
1	1	1	0	Setting prohibited	40.125 μs	20.063 μs	16.05 μs	13.376 μs
1	1	1	1	Setting prohibited	43.125 μs	21.563 μs	17.25 μs	14.375 μs

(11) Port mode registers 8 to 10 (PM8 to PM10)

When using the ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97, and ANI16/P100 to ANI23/P107 pins for analog input port, set PM8_0 to PM8_7, PM9_0 to PM9_7, and PM10_0 to PM10_7 to 1. The output latches of P8_0 to P8_7, P9_0 to P9_7, and P10_0 to P10_7 at this time may be 0 or 1.

If PM8_0 to PM8_7, PM9_0 to PM9_7, and PM10_0 to PM10_7 are set to 0, they cannot be used as analog input port pins.

PM8 to PM10 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Caution If a pin is set as an analog input port, not the pin level but “0” is always read.

<R> **Figure 10-14. Formats of Port Mode Registers 8 to 10 (PM8 to PM10)**

Address: FFF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	PM8_7	PM8_6	PM8_5	PM8_4	PM8_3	PM8_2	PM8_1	PM8_0

Address: FFF29H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	PM9_7	PM9_6	PM9_5	PM9_4	PM9_3	PM9_2	PM9_1	PM9_0

Address: FFF2AH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM10	PM10_7	PM10_6	PM10_5	PM10_4	PM10_3	PM10_2	PM10_1	PM10_0

PMm_n	Pmn pin I/O mode selection (mn = 80 to 87, 90 to 97, 100 to 107)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

The ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/PM97, and ANI16/P100 to ANI23/P107 pins are as shown below depending on the settings of PM8 to PM10.

Remark ANI00/P80 to ANI07/P87, ANI08/P90 to ANI10/P92: 78K0R/HC3
ANI00/P80 to ANI07/P87, ANI08/P90 to ANI14/P96: 78K0R/HE3
ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97: 78K0R/HF3
ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97, ANI16/P100 to ANI23/P107: 78K0R/HG3

Table 10-10. Setting Functions of ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97, and ANI16/P100 to ANI23/P107 Pins

ADPC	PM8 to PM10	ADS	ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97, and ANI16/P100 to ANI23/P107 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

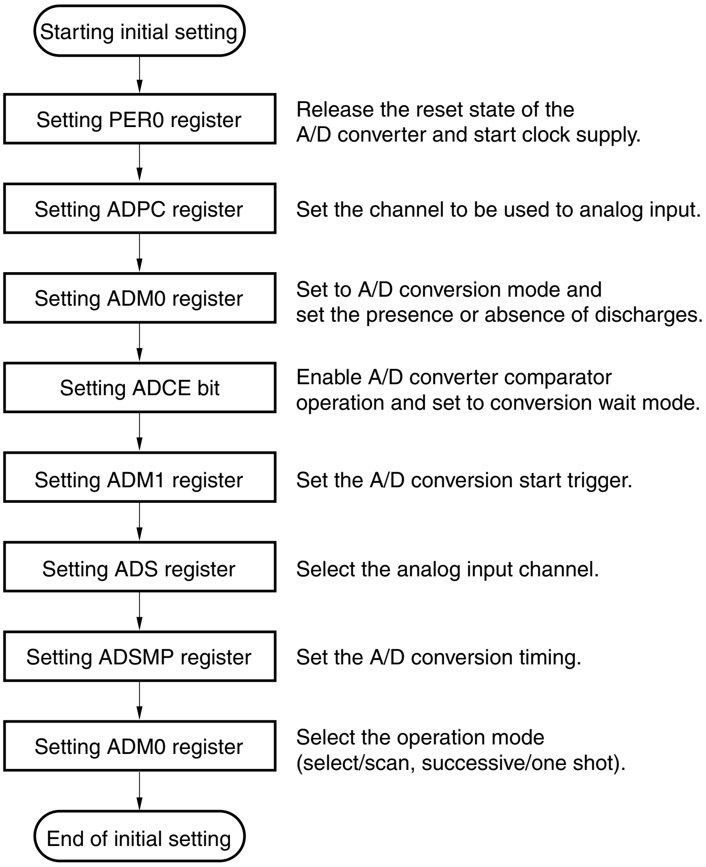
Remark ANI00/P80 to ANI07/P87, ANI08/P90 to ANI10/P92: 78K0R/HC3
ANI00/P80 to ANI07/P87, ANI08/P90 to ANI14/P96: 78K0R/HE3
ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97: 78K0R/HF3
ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97, ANI16/P100 to ANI23/P107: 78K0R/HG3

10.4 A/D Converter Operations

10.4.1 Basic operations of A/D converter

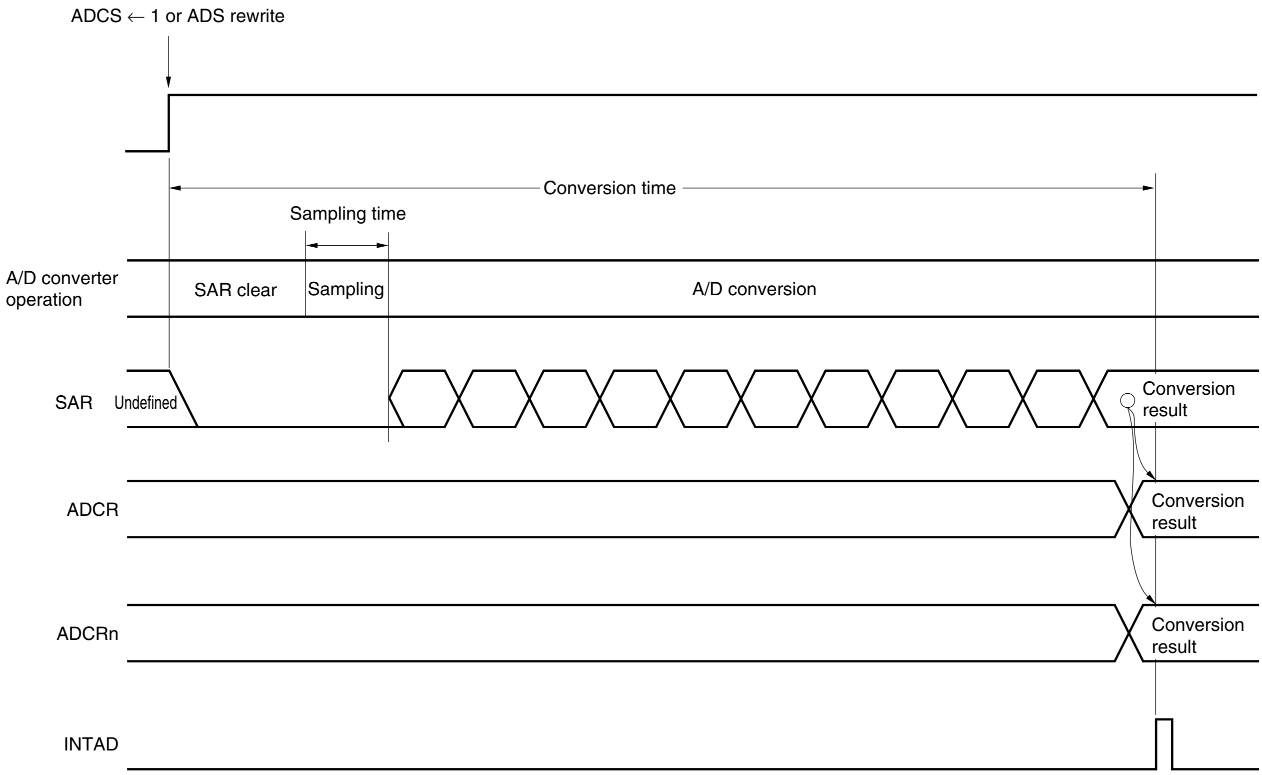
The initial A/D conversion setting procedure is shown below.

Figure 10-15. Initial A/D Converter Setting Flow


Starting initial setting
Setting PER0 register
Release the reset state of the A/D converter and start clock supply.
Setting ADPC register
Set the channel to be used to analog input.
Setting ADM0 register
Set to A/D conversion mode and set the presence or absence of discharges.
Setting ADCE bit
Enable A/D converter comparator operation and set to conversion wait mode.
Setting ADM1 register
Set the A/D conversion start trigger.
Setting ADS register
Select the analog input channel.
Setting ADSMP register
Set the A/D conversion timing.
Setting ADM0 register
Select the operation mode (select/scan, successive/one shot).
End of initial setting

Caution Make sure the period of conversion wait time is 1 μs or more.

Figure 10-16. Basic Operation of A/D Converter (Continuous Select Mode)



A/D conversion operations are performed continuously until bit 7 (ADCS) of A/D converter mode register 0 (ADM0) is reset (0) by software.

If a write operation is performed to the analog input channel specification register (ADS) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (1), conversion starts again from the beginning.

Reset signal generation clears the A/D conversion result common register (ADCR, ADCRH) and A/D conversion result register n (ADCRn, ADCRnH) to 0000H or 00H.

Remark n = 0 to 10: 78K0R/HC3
n = 0 to 14: 78K0R/HE3
n = 0 to 15: 78K0R/HF3
n = 0 to 23: 78K0R/HG3

10.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI00 to ANI23) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result common register (ADCR) and10-bit A/D conversion result register n (ADCRn)) is shown by the following expression.

$$\mathrm{SAR} = \mathrm{INT}\ (\frac{V_{\mathrm{AIN}}}{AV_{\mathrm{REF}}} \times 1024 + 0.5)$$

$$\mathrm{ADCR} = \mathrm{SAR} \times 64$$

or

$$(\frac{\mathrm{ADCR}}{64} - 0.5) \times \frac{AV_{\mathrm{REF}}}{1024} \leq V_{\mathrm{AIN}} < (\frac{\mathrm{ADCR}}{64} + 0.5) \times \frac{AV_{\mathrm{REF}}}{1024}$$

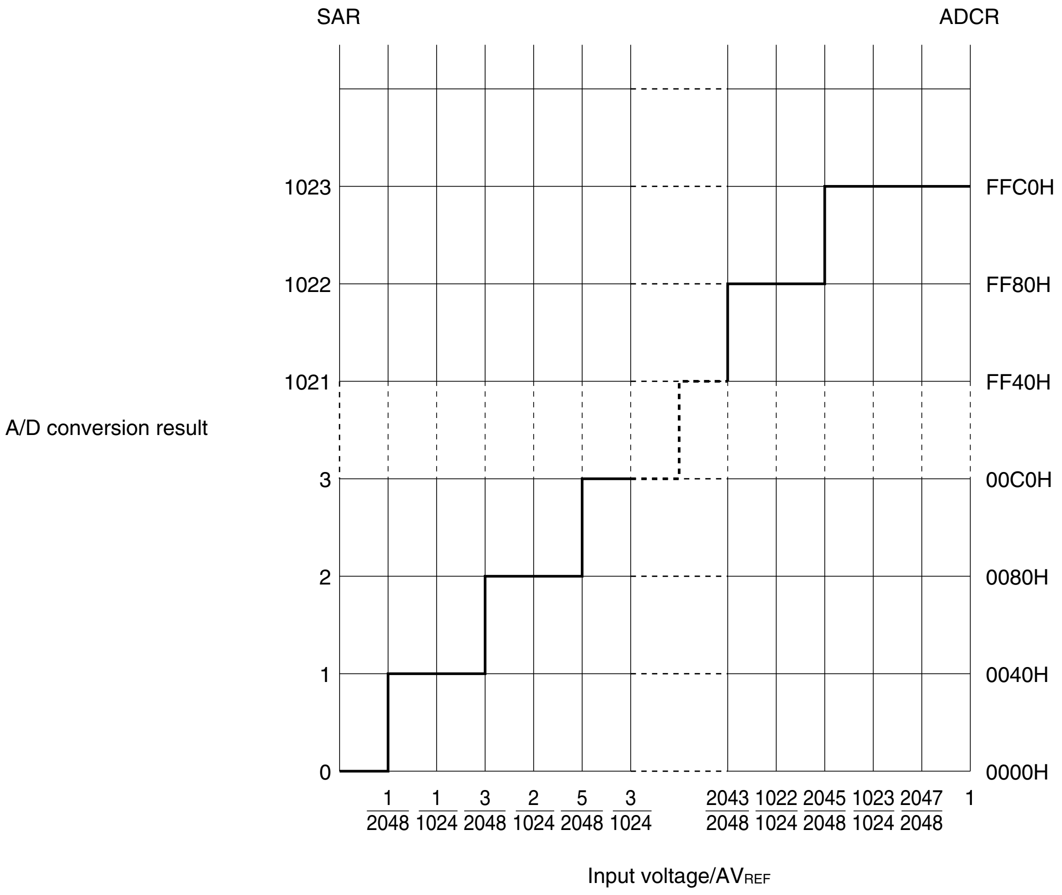
where, INT( ): Function which returns integer part of value in parentheses
V_{AIN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR: A/D conversion result common register (ADCR) value
SAR: Successive approximation register

Remarks 1. Substitute ADCRn for ADCR in the expression to find how ADCRn and the analog input voltage relate.

2. n = 0 to 10, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI10/P92: 78K0R/HC3
n = 0 to 14, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI14/P96: 78K0R/HE3
n = 0 to 15, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97: 78K0R/HF3
n = 0 to 23, ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97, ANI16/P100 to ANI23/P107: 78K0R/HG3

Figure 10-17 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 10-17. Relationship Between Analog Input Voltage and A/D Conversion Result



10.4.3 Trigger mode selection

Software trigger mode and hardware trigger mode are provided as the trigger modes that set the A/D conversion start timing. Timer trigger modes 0 to 1 and external trigger mode are provided as hardware trigger modes. These trigger modes are set by using the ADM1 register.

- Software trigger mode
- Hardware trigger mode (timer trigger modes 0 to 1, external trigger mode)

(1) Software trigger mode

By setting ADCS = 1, this mode starts A/D conversion of the analog input channels (ANI00 to ANI23) selected by the analog input channel specification register (ADS).

After A/D conversion ends, A/D conversion is successively repeated as long as ADCS bit = 0 is not set.

A/D conversion is aborted if the ADM0, ADM1, ADS, and ADPC registers are written during a conversion operation. In such a case, A/D conversion is performed again from the start in select mode, and A/D conversion is restarted from scan 0 in scan mode.

(2) Hardware trigger mode

(a) External trigger mode

A conversion operation is started by inputting an external trigger (ADTRG pin) to the analog inputs (ANI00 to ANI23 pins) specified by the ADS register. The edge detection (rising edge, falling edge, both rising and falling edges) of the external trigger can be specified by setting the ADETS1 and ADETS0 bits of the ADM1 register. When the ADCS and ADCE bits of the ADM0 register are set (1), a trigger wait state is entered and conversion is started after an external trigger is input.

When conversion ends, the conversion result is stored in the ADCR and ADCRn registers.

When ADMD0 = 0, an A/D conversion end interrupt request signal (INTAD) is generated at the same time as when the conversion result is stored in the ADCR and ADCRn registers and the trigger wait state is entered again. When ADMD0 = 1, an A/D conversion end interrupt request signal (INTAD) is generated after the conversion of the channel specified using the ADS register ends and the trigger wait state is entered again.

If the ADM0, ADM1, ADS, and ADPC registers are written during a conversion operation, conversion is aborted and the trigger wait state is entered again.

Caution Before changing the trigger mode, to clear the ADCE bit to 0.

Remark n = 0 to 10, ANI00 to ANI10: 78K0R/HC3
n = 0 to 14, ANI00 to ANI14: 78K0R/HE3
n = 0 to 15, ANI00 to ANI15: 78K0R/HF3
n = 0 to 23, ANI00 to ANI23: 78K0R/HG3

(b) Timer trigger mode

The generation of a timer interrupt request signal (INTTM12, INTTM22) in this mode starts a conversion operation for the analog inputs (ANI00 to ANI23 pins) specified by the ADS register. According to the settings of the ADTMS1 and ADTMS0 bits of the ADM1 register, a timer interrupt request signal (INTTAM12, INTTM22) is specified and conversion is started at the rising edge of the specified interrupt request signal. When the ADCS and ADCE bits of the ADM0 register are set (1), a trigger wait state is entered and conversion is started after a timer interrupt signal is input.

When conversion ends, the conversion result is stored in the ADCR and ADCRn registers.

When ADMD0 = 0, an A/D conversion end interrupt request signal (INTAD) is generated at the same time as when the conversion result is stored in the ADCR and ADCRn registers and the trigger wait state is entered again. When ADMD0 = 1, an A/D conversion end interrupt request signal (INTAD) is generated after the conversion of the channel specified using the ADS register ends and the trigger wait state is entered again.

When a valid trigger is input during the conversion operation, conversion is aborted and performed again from the start.

If the ADM0, ADM1, ADS, and ADPC registers are written during a conversion operation, conversion is aborted and the trigger wait state is entered again.

Caution Before changing the trigger mode, to clear the ADCE bit to 0.

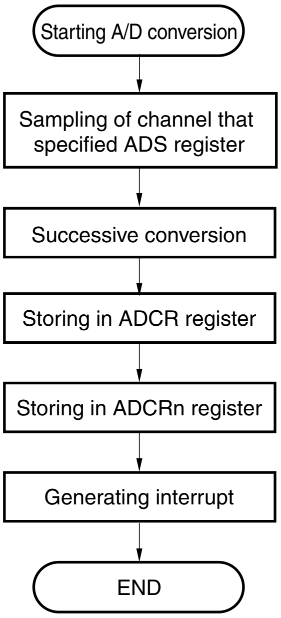
Remark n = 0 to 10, ANI00 to ANI10: 78K0R/HC3
n = 0 to 14, ANI00 to ANI14: 78K0R/HE3
n = 0 to 15, ANI00 to ANI15: 78K0R/HF3
n = 0 to 23, ANI00 to ANI23: 78K0R/HG3

10.4.4 A/D converter operation modes

Four A/D converter operation modes are provided, namely continuous select mode, continuous scan mode, one-shot select mode, and one-shot scan mode are provided.

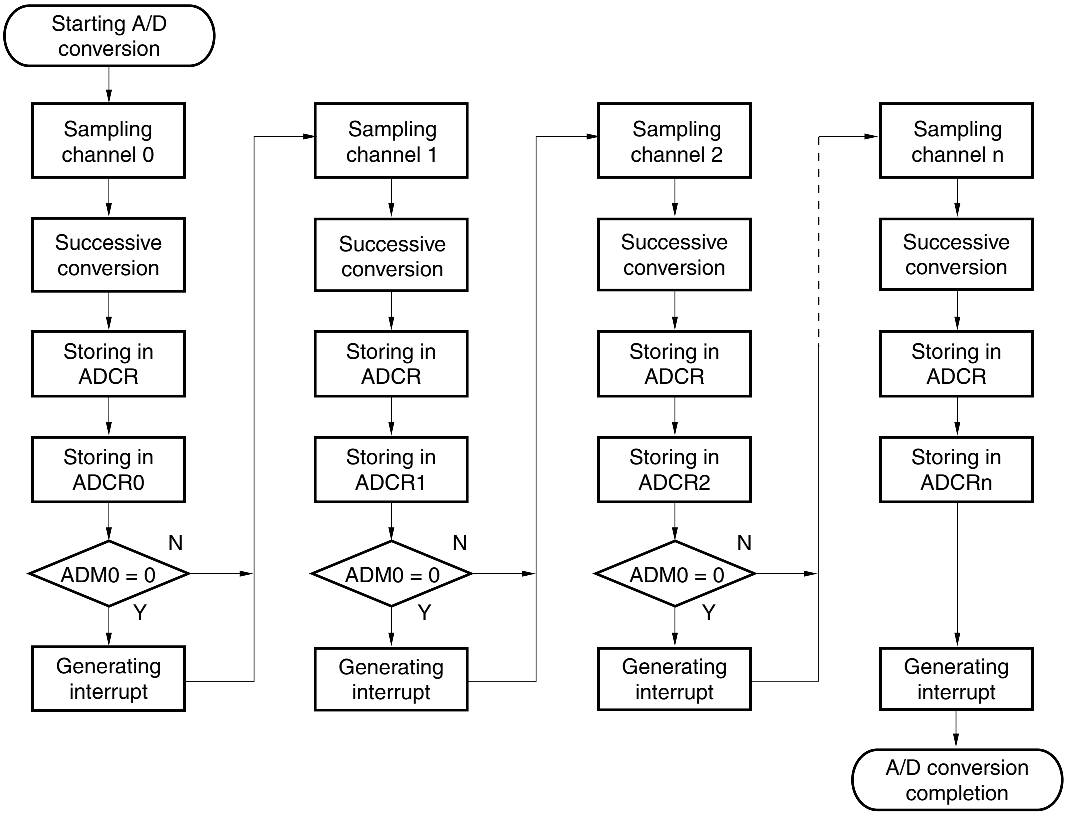
The operation mode is set by using the ADMD0 to ADMD2 bits of the ADM0 register.

Figure 10-18. Select Mode Operation Flow



Remark n = 0 to 10: 78K0R/HC3
n = 0 to 14: 78K0R/HE3
n = 0 to 15: 78K0R/HF3
n = 0 to 23: 78K0R/HG3

Figure 10-19. Scan Mode Operation Flow

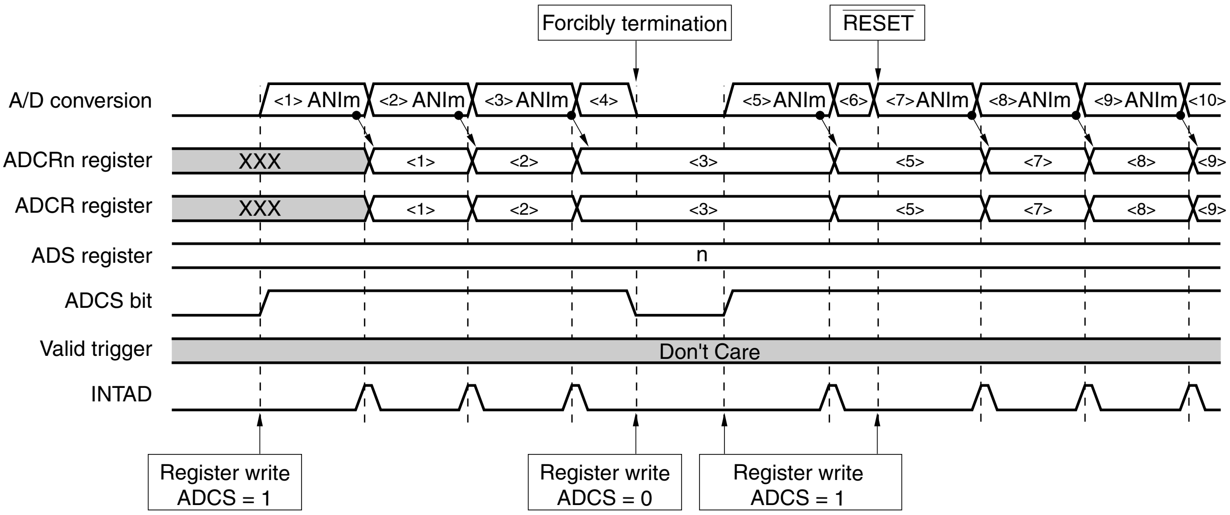


Remark n = 0 to 10: 78K0R/HC3
n = 0 to 14: 78K0R/HE3
n = 0 to 15: 78K0R/HF3
n = 0 to 23: 78K0R/HG3

(1) Continuous select mode

This mode successively performs A/D conversion of the voltage of one analog input pin specified by the ADS register. The conversion result is stored in the ADCR register, and the ADCRn register that corresponds to the analog input pin. In this mode, the analog input pins and ADCRn registers have a one-to-one correspondence, and an A/D conversion end interrupt request signal (INTAD) is generated at the end of each A/D conversion. After the end of conversion, the next conversion is repeatedly performed as long as the ADCS bit of the ADM0 register is not set to “0”.

Figure 10-20. Continuous Select Mode Operation Timing Example (Software Trigger Mode)



Remarks 1. A/D conversion is stopped by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 0).

A/D conversion is started after A/D conversion is reset by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 1).

A valid trigger is not detected.

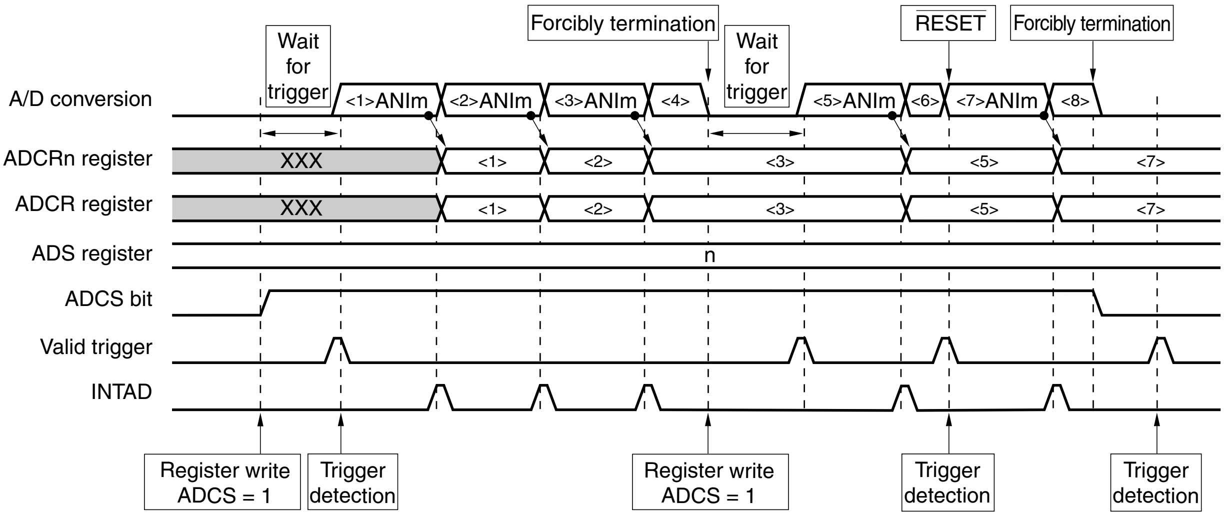
2. n = 0 to 10, m = 00 to 10: 78K0R/HC3

n = 0 to 14, m = 00 to 14: 78K0R/HE3

n = 0 to 15, m = 00 to 15: 78K0R/HF3

n = 0 to 23, m = 00 to 23: 78K0R/HG3

Figure 10-21. Continuous Select Mode Operation Timing Example (Hardware Trigger Mode)



Remarks 1. A/D conversion is stopped by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 0).

A/D conversion is started after A/D conversion is reset by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 1).

A valid trigger is not detected.

2. n = 0 to 10, m = 00 to 10: 78K0R/HC3
n = 0 to 14, m = 00 to 14: 78K0R/HE3
n = 0 to 15, m = 00 to 15: 78K0R/HF3
n = 0 to 23, m = 00 to 23: 78K0R/HG3

(2) Continuous scan mode

This mode sequentially selects pins from the ANI00 pin to the analog input pin specified by the ADS register, and performs A/D conversion successively.

The conversion result is stored in the ADCR register and the ADCRn register that corresponds to the analog input pin.

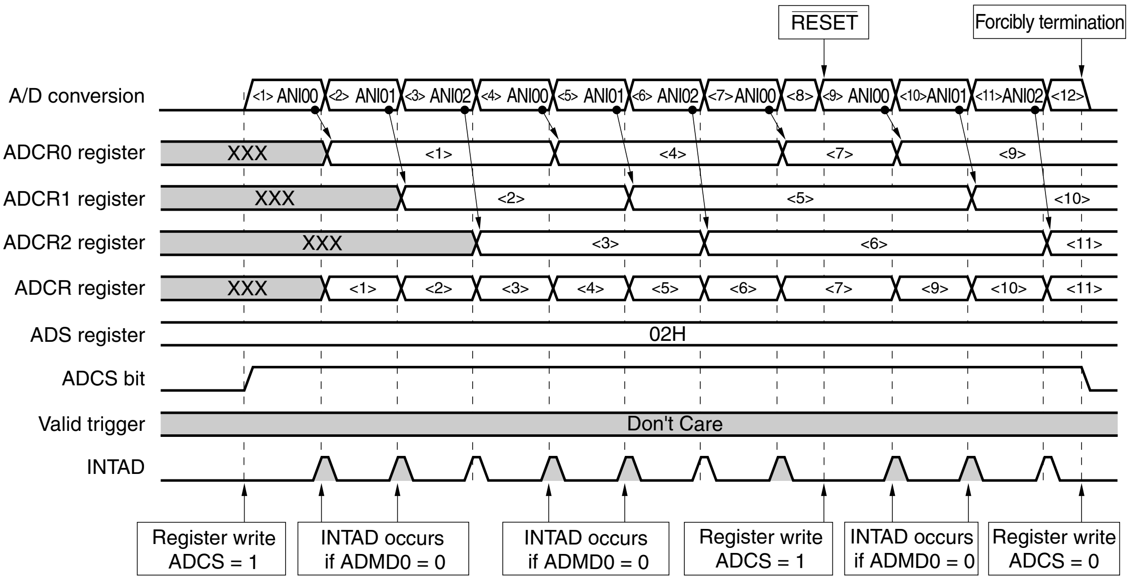
When ADMD0 = 0, an A/D conversion end interrupt request signal (INTAD) is generated for each A/D conversion.

When converting the analog input pin specified by the ADS register ends while ADMD0 = 1, an A/D conversion end interrupt request signal (INTAD) is generated, and as long as ADCS = 0 is not set, A/D conversion is started from the ANI00 pin again.

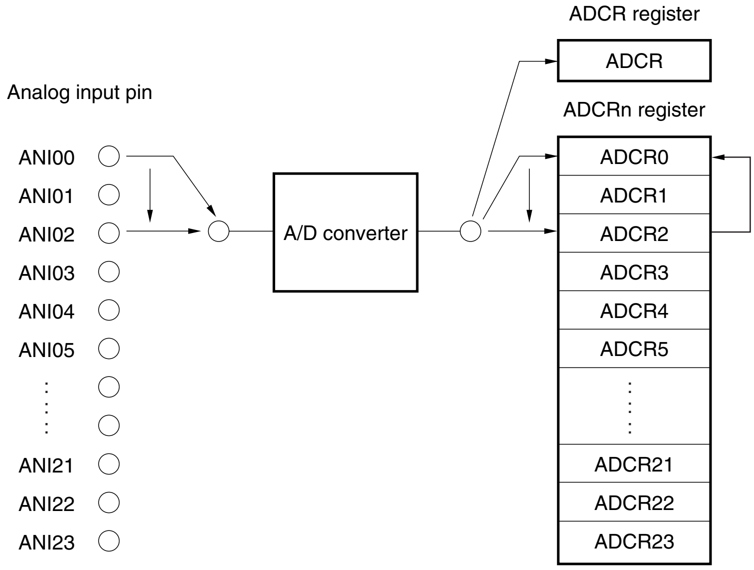
Remark n = 0 to 10: 78K0R/HC3
n = 0 to 14: 78K0R/HE3
n = 0 to 15: 78K0R/HF3
n = 0 to 23: 78K0R/HG3

Figure 10-22. Continuous Scan Mode Operation Timing Example (ADS Register = 02H, Software Trigger Mode)

(a) Timing example



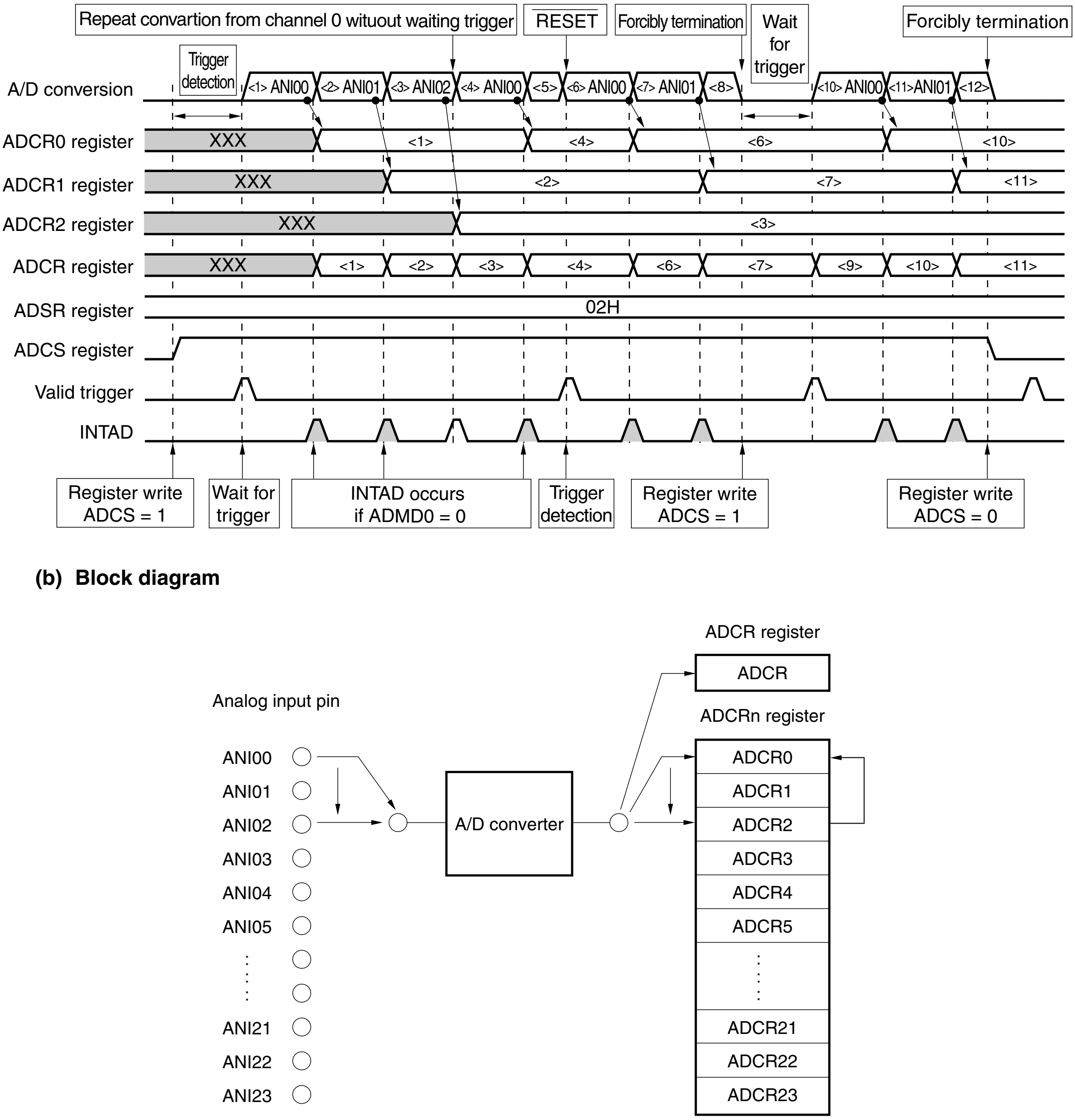
(b) Block diagram



Remarks 1. A/D conversion is stopped by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 0).
A/D conversion is started after A/D conversion is reset by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 1).
A valid trigger is not detected.

2. n = 0 to 10, ANI00 to ANI10: 78K0R/HC3
n = 0 to 14, ANI00 to ANI14: 78K0R/HE3
n = 0 to 15, ANI00 to ANI15: 78K0R/HF3
n = 0 to 23, ANI00 to ANI23: 78K0R/HG3

Figure 10-23. Continuous Scan Mode Operation Timing Example (ADS Register = 02H, Hardware Trigger Mode)



(a) Timing example

(b) Block diagram

Remarks 1. A/D conversion is stopped by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 0).

A/D conversion is started after A/D conversion is reset by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 1).

A valid trigger is not detected.

2. n = 0 to 10, ANI00 to ANI10: 78K0R/HC3

n = 0 to 14, ANI00 to ANI14: 78K0R/HE3

n = 0 to 15, ANI00 to ANI15: 78K0R/HF3

n = 0 to 23, ANI00 to ANI23: 78K0R/HG3

(3) One-shot select mode

This mode performs a single A/D conversion of one analog input pin specified by the ADS register.

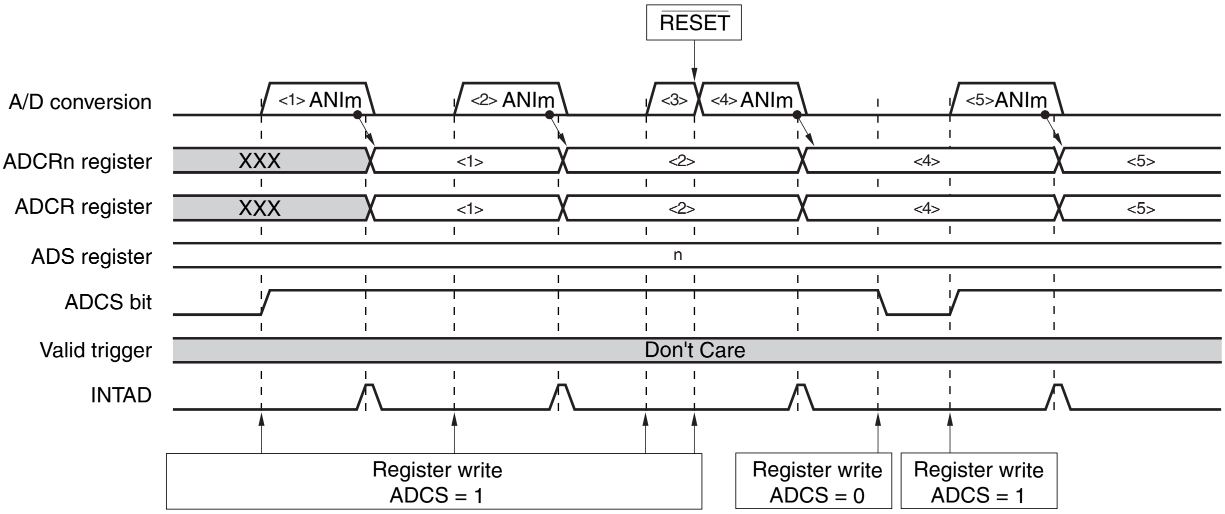
The conversion result is stored in the ADCRn register corresponding to the analog input pin. In this mode, the analog input pins and ADCRn registers have a one-to-one correspondence, and an A/D conversion end interrupt request signal (INTAD) is generated at the end of single A/D conversion. After the end of A/D conversion, A/D conversion operation is stopped.

After the end of conversion in software trigger mode, the next conversion is started by setting ADCS = 1.

After the end of conversion in hardware trigger mode, a trigger wait state is entered.

Remark n = 0 to 10: 78K0R/HC3
n = 0 to 14: 78K0R/HE3
n = 0 to 15: 78K0R/HF3
n = 0 to 23: 78K0R/HG3

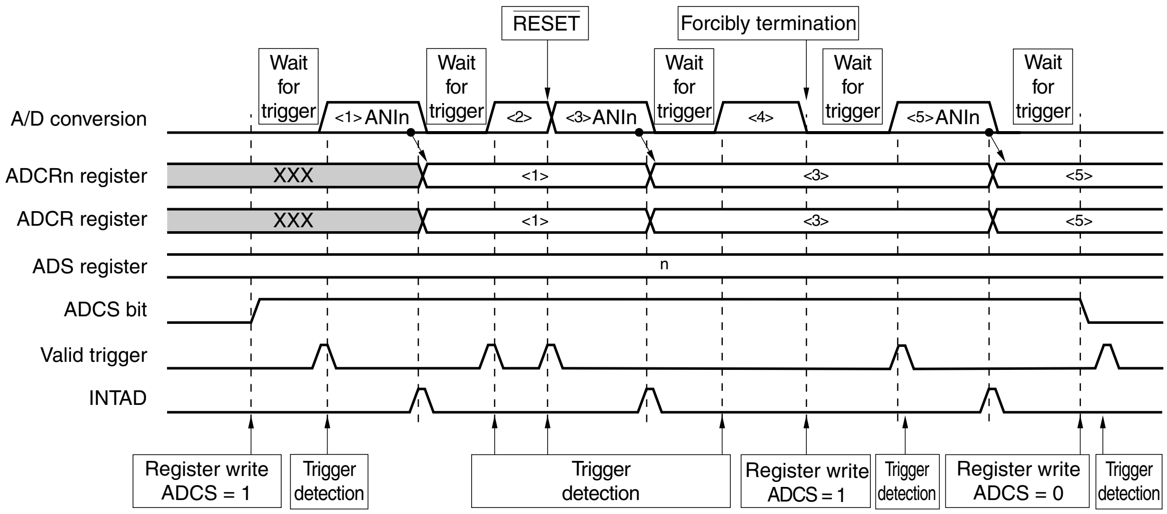
<R> **Figure 10-24. One-Shot Select Mode Operation Timing Example (Software Trigger Mode)**



Remarks 1. A/D conversion is stopped by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 0).
A/D conversion is started after A/D conversion is reset by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 1).
A valid trigger is not detected.

2. n = 0 to 10, m = 00 to 10: 78K0R/HC3
n = 0 to 14, m = 00 to 14: 78K0R/HE3
n = 0 to 15, m = 00 to 15: 78K0R/HF3
n = 0 to 23, m = 00 to 23: 78K0R/HG3

Figure 10-25. One-Shot Select Mode Operation Timing Example (Hardware Trigger Mode)



Remarks 1. A/D conversion is stopped by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 0).
A/D conversion is started after A/D conversion is reset by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 1).
A valid trigger is not detected.

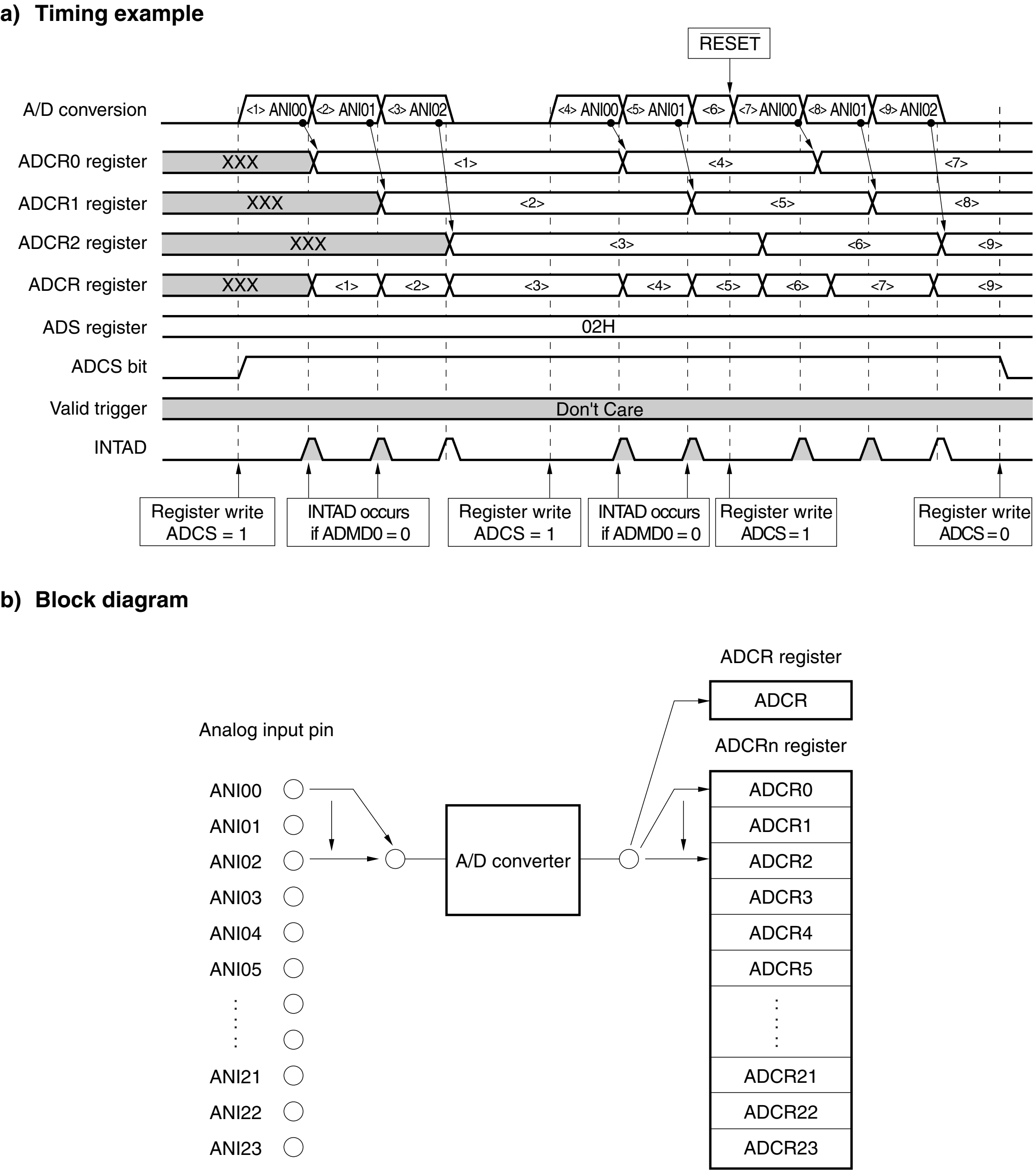
2. n = 0 to 10, m = 00 to 10: 78K0R/HC3
n = 0 to 14, m = 00 to 14: 78K0R/HE3
n = 0 to 15, m = 00 to 15: 78K0R/HF3
n = 0 to 23, m = 00 to 23: 78K0R/HG3

(4) One-shot scan mode

This mode sequentially selects pins from the ANI00 pin to the analog input pin specified by the ADS register, and performs A/D conversion.
The conversion result is stored in the ADCR register and the ADCRn register that corresponds to the analog input pin.
When conversion up to the analog input pin specified by the ADS register is complete, a stop state is entered.
When ADMD0 = 0, an A/D conversion end interrupt request signal (INTAD) is generated for each A/D conversion.
When converting the analog input pin specified by the ADS register ends while ADMD0 = 1, an A/D conversion end interrupt request signal (INTAD) is generated.
After the end of conversion in software trigger mode, the next conversion is started by setting ADCS = 1.
After the end of conversion in hardware trigger mode, a trigger wait state is entered.

Remark n = 0 to 10: 78K0R/HC3
n = 0 to 14: 78K0R/HE3
n = 0 to 15: 78K0R/HF3
n = 0 to 23: 78K0R/HG3

Figure 10-26. One-Shot Select Mode Operation Timing Example (ADS Register = 02H, Software Trigger Mode)



(a) Timing example

(b) Block diagram

Remarks 1. A/D conversion is stopped by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 0).

A/D conversion is started after A/D conversion is reset by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 1).

A valid trigger is not detected.

2. n = 0 to 10, m = 00 to 10: 78K0R/HC3

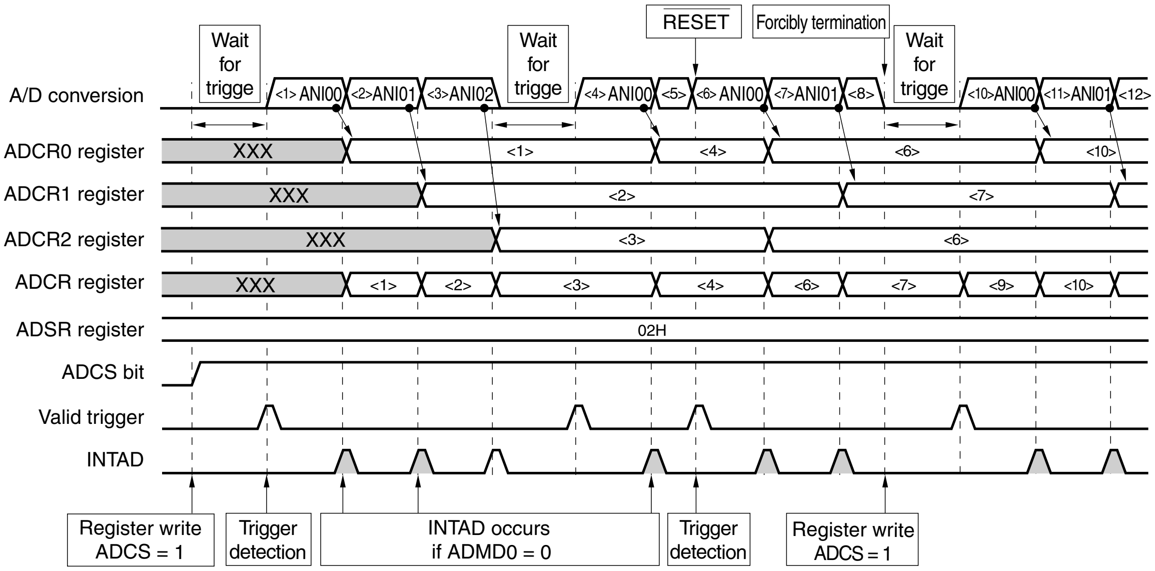
n = 0 to 14, m = 00 to 14: 78K0R/HE3

n = 0 to 15, m = 00 to 15: 78K0R/HF3

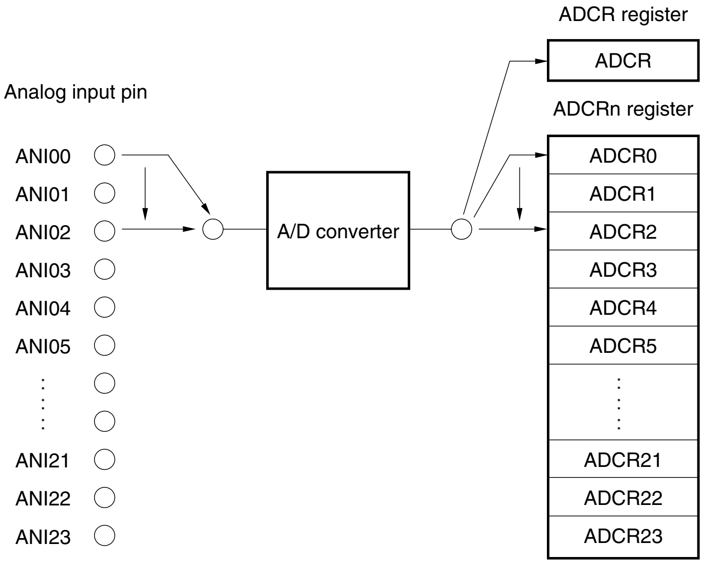
n = 0 to 23, m = 00 to 23: 78K0R/HG3

Figure 10-27. One-Shot Select Mode Operation Timing Example (ADS Register = 02H, Hardware Trigger Mode)

(a) Timing example



(b) Block diagram



Remarks 1. A/D conversion is stopped by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 0).

A/D conversion is started after A/D conversion is reset by writing to the ADM0, ADM1, ADS, ADS, ADCR (ADCRH), or ADPC register (ADCS = 1).

A valid trigger is not detected.

2. n = 0 to 10, ANI00 to ANI10: 78K0R/HC3

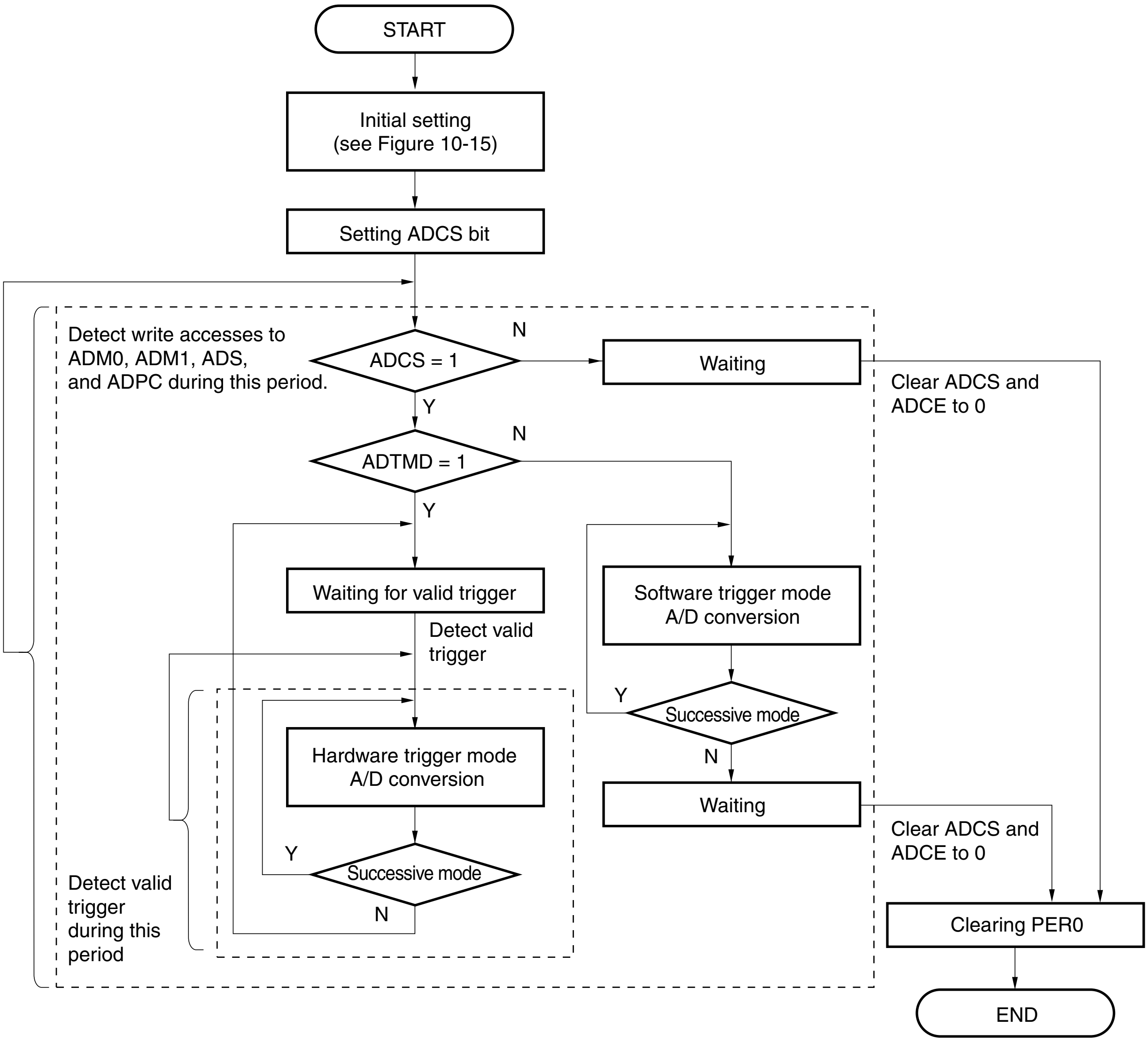
n = 0 to 14, ANI00 to ANI14: 78K0R/HE3

n = 0 to 15, ANI00 to ANI15: 78K0R/HF3

n = 0 to 23, ANI00 to ANI23: 78K0R/HG3

The following figure describes the setting method.

Figure 10-28. A/D Conversion Operation Flow



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.

$$1\text{LSB} = 1/2^{10} = 1/1024$$
$$= 0.098\%\text{FSR}$$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±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 10-29. Overall Error

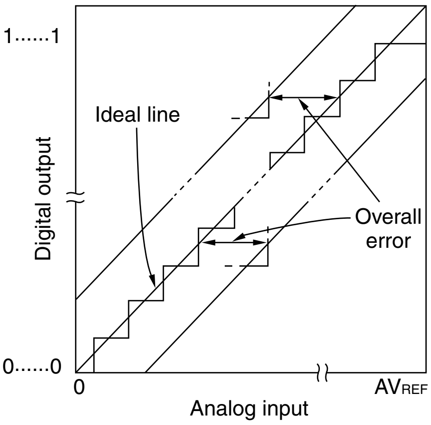
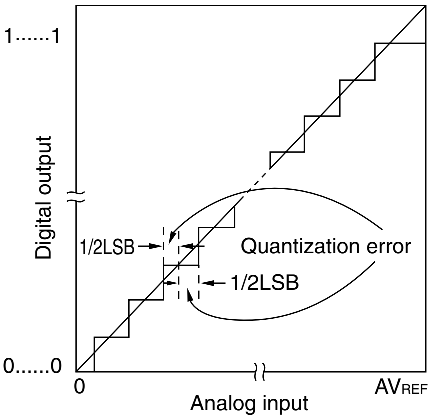


Figure 10-30. Quantization Error



(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 10-31. Zero-Scale Error

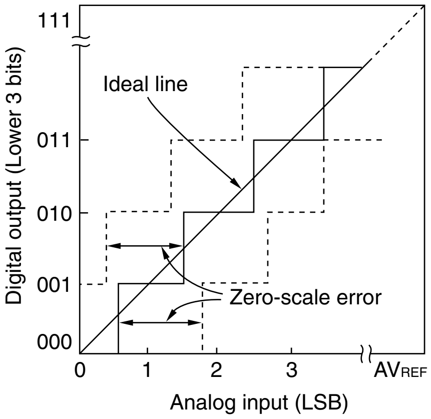


Figure 10-32. Full-Scale Error

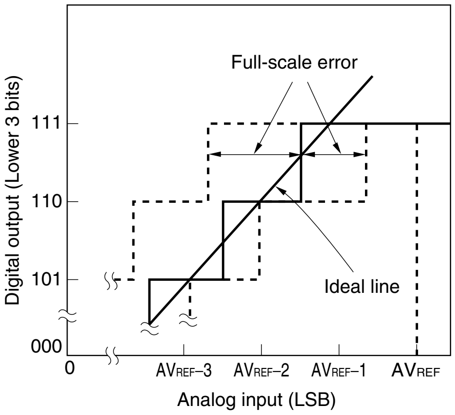


Figure 10-33. Integral Linearity Error

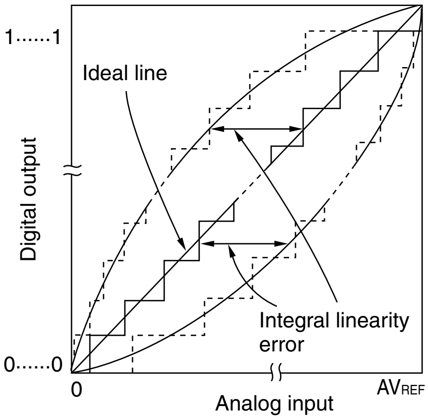
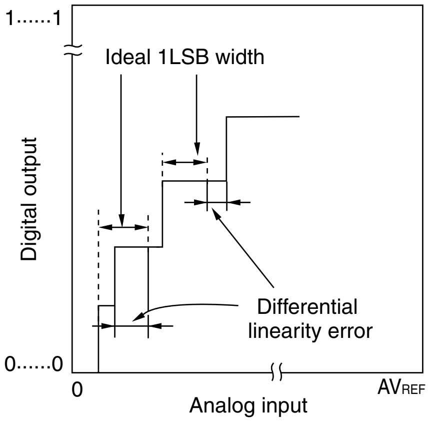


Figure 10-34. Differential Linearity Error



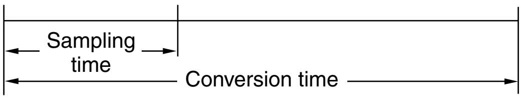
(8) Conversion time

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 A/D Converter

(1) Operating current in STOP mode

Shift to STOP mode after clearing the A/D converter (by clearing bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by clearing bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1H (IF1H) to 0 and start operation.

(2) Reducing current when A/D converter is stopped

If bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register 0 (ADM0) are set to 0, the current will not be increased by the A/D converter even if a voltage is applied to AV_{REF}, while the A/D converter is stopped.

(3) Input range of ANI00 to ANI23

Observe the rated range of the ANI00 to ANI23 input voltage. If a voltage of AV_{REF} or higher and AV_{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.

(4) Conflicting operations

<1> Conflict between A/D conversion result common register (ADCR, ADCRH) or A/D conversion result register n (ADCRn, ADCRnH) write and ADCR, ADCRH or ADCRn, ADCRnH read by instruction upon the end of conversion
ADCR, ADCRH or ADCRn, ADCRnH read has priority. After the read operation, the new conversion result is written to ADCR or ADCRH, ADCRn or ADCRnH.

<2> Conflict between ADCR, ADCRH or ADCRn, ADCRnH write and A/D converter mode register 0, 1 (ADM0, ADM1) write, analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion
ADM0, ADM1, ADS, or ADPC write has priority. ADCR, ADCRH, ADCRn, or ADCRnH write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

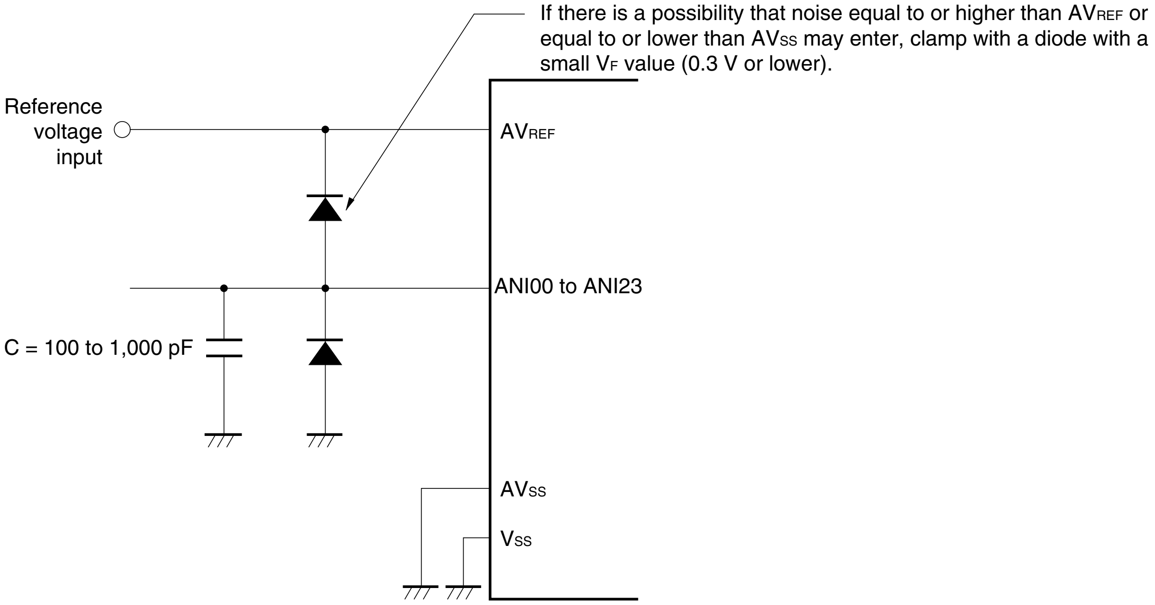
(5) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REF} pin and pins ANI00 to ANI23.

<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 10-35 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.

Remark n = 0 to 10, ANI00 to ANI10: 78K0R/HC3
n = 0 to 14, ANI00 to ANI14: 78K0R/HE3
n = 0 to 15, ANI00 to ANI15: 78K0R/HF3
n = 0 to 23, ANI00 to ANI23: 78K0R/HG3

Figure 10-35. Analog Input Pin Connection


If there is a possibility that noise equal to or higher than AVREF or equal to or lower than AVSS may enter, clamp with a diode with a small VF value (0.3 V or lower).
Reference voltage input
AVREF
ANI00 to ANI23
C = 100 to 1,000 pF
AVSS
VSS

(6) ANI00/P80 to ANI07/P87 and ANI08/P90 to ANI15/P97 and ANI16/P100 to ANI23/P107

<1> The analog input pins (ANI00 to ANI07) are also used as input port pins (P80 to P87).
The analog input pins (ANI08 to ANI15) are also used as input port pins (P90 to P97).
The analog input pins (ANI16 to ANI23) are also used as input port pins (P100 to P107).
When A/D conversion is performed with any of ANI00 to ANI23 selected, do not access P80 to P87 and P90 to P97 and P100 to P107 while conversion is in progress; otherwise the conversion resolution may be degraded.
It is recommended to select the pins to be used as P80 to P87, P90 to P97, and P100 to P107, from the analog input pins that are furthest from AV_{REF}.

<2> If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.

(7) Input impedance of ANI00 to ANI23 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 ANI00 to ANI23 pins (see **Figure 10-35**).

Remark ANI00/P80 to ANI07/P87, ANI08/P90 to ANI10/P92: 78K0R/HC3
ANI00/P80 to ANI07/P87, ANI08/P90 to ANI14/P96: 78K0R/HE3
ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97: 78K0R/HF3
ANI00/P80 to ANI07/P87, ANI08/P90 to ANI15/P97, ANI16/P100 to ANI23/P107: 78K0R/HG3

(8) AV$_{REF}$ pin input impedance

A series resistor string of several tens of kΩ is connected between the AV$_{REF}$ and AV$_{SS}$ pins.

Therefore, if the output impedance of the reference voltage source is high, this will result in a series connection to the series resistor string between the AV$_{REF}$ and AV$_{SS}$ pins, resulting in a large reference voltage error.

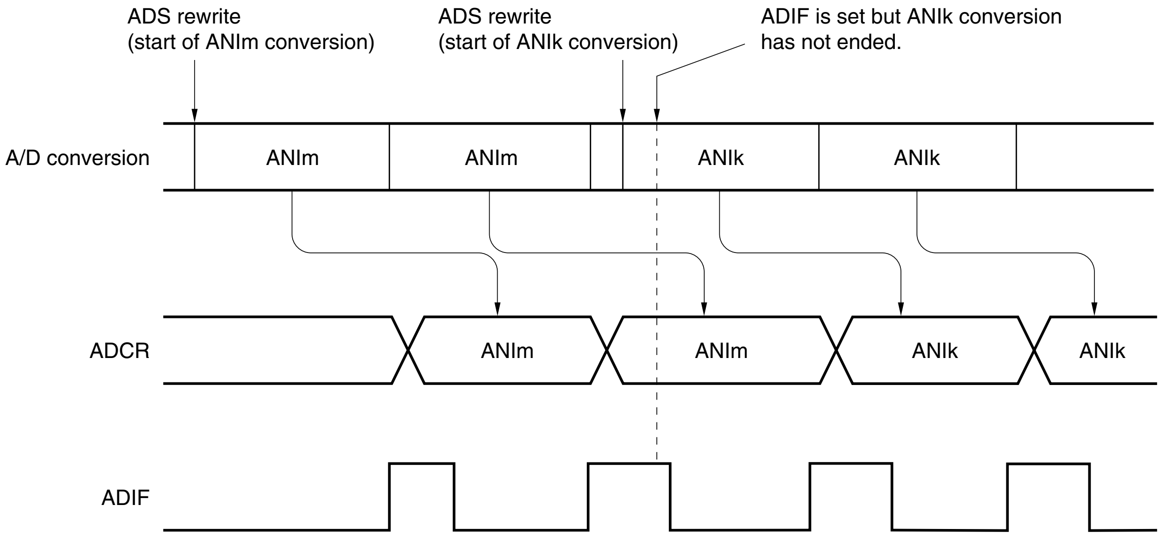
(9) Interrupt request flag (ADIF)

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 10-36. Timing of A/D Conversion End Interrupt Request Generation


ADS rewrite
(start of ANIm conversion)
ADS rewrite
(start of ANIk conversion)
ADIF is set but ANIk conversion has not ended.
A/D conversion
ANIm
ANIm
ANIk
ANIk
ADCR
ANIm
ANIm
ANIk
ANIk
ADIF

Remarks 1. m = 00 to 10: 78K0R/HC3
m = 00 to 14: 78K0R/HE3
m = 00 to 15: 78K0R/HF3
m = 00 to 23: 78K0R/HG3

2. k = 00 to 10: 78K0R/HC3
k = 00 to 14: 78K0R/HE3
k = 00 to 15: 78K0R/HF3
k = 00 to 23: 78K0R/HG3

(10) 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. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(11) A/D conversion common result register (ADCR, ADCRH) or A/D conversion result register n (ADCRn)read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), 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 ADM0, ADM1, ADS, or ADPC. Using a timing other than the above may cause an incorrect conversion result to be read.

(12) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 10-37. Internal Equivalent Circuit of ANIm Pin

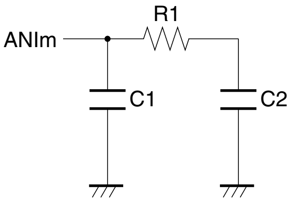


Table 10-11. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	Mode	R1	C1	C2
4.0 V ≤ V_{DD} ≤ 5.5 V	Normal	5.2 kΩ	8 pF	6.3 pF
	High speed 1	5.2 kΩ		
	High speed 2	7.8 kΩ		
2.7 V ≤ V_{DD} < 4.0 V	Normal/High speed 1	18.6 kΩ		
	High speed 2	7.8 kΩ		

Remarks 1. The resistance and capacitance values shown in Table 10-6 are not guaranteed values.

2. m = 00 to 10: 78K0R/HC3
m = 00 to 14: 78K0R/HE3
m = 00 to 15: 78K0R/HF3
m = 00 to 23: 78K0R/HG3

(13) Starting the A/D converter

Start the A/D converter after the AV_{REF} voltage stabilizes.

CHAPTER 11 SERIAL ARRAY UNIT

Product SAU	78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
	yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
CSI	2	3		4
UART	–	1		
Simplified I^2C	1	2		

The serial array unit has two 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/Hx3 is as shown below.

Table 11-1. Serial Function Assignment of Each Product

HC3	HE3	HF3	HG3	Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
√	√	√	√	0	0	CSI00 (supports SPI)	–	–
–	√	√	√		1	CSI01(supports SPI)		–
√	√	√	√	1	0	CSI10	–	–
Note	**Note**	**Note**	√		1	CSI11		IIC11
–	√	√	√	2	0	–	UART2	IIC20
					1	–		–

Note Only IIC11 is mounted.

Remarks 1. When using CSI11 in channel 1 of unit 1, IIC11 cannot be used.
When using IIC11, CSI11 cannot be used.

2. When using UART2 in channel 0 or 1 of unit 2, IIC20 cannot be used.
When using IIC20, UART2 cannot be used.

The descriptions in this chapter are for the 78K0R/HG3.

11.1 Functions of Serial Array Unit

Each serial interface supported by the 78K0R/Hx3 has the following features.

11.1.1 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11)

This is a clocked communication function that uses three lines: serial clock ($\overline{\text{SCK}}$) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 to 16 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

Furthermore, CSI00 and CSI01 (channels 0 and 1 of unit 0) support the SPI function.

[Expansion function]

- Slave select function of the SPI function

11.1.2 UART (UART2)

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 7 to 9, or 16 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 detection flag]

- Framing error, parity error, or overrun error

11.1.3 Simplified I^{2}C (IIC11, 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 I^{2}C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master 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]

- Overrun error
- Parity error (ACK error)

* [Functions not supported by simplified I^{2}C]

- 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 SOEm_n (SOEm register) bit and stopping the output of serial communication data. See **11.8.3 (2) Processing flow** for details.

Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1)

11.2 Configuration of Serial Array Unit

Serial array unit includes the following hardware.

Table 11-2. Configuration of Serial Array Unit

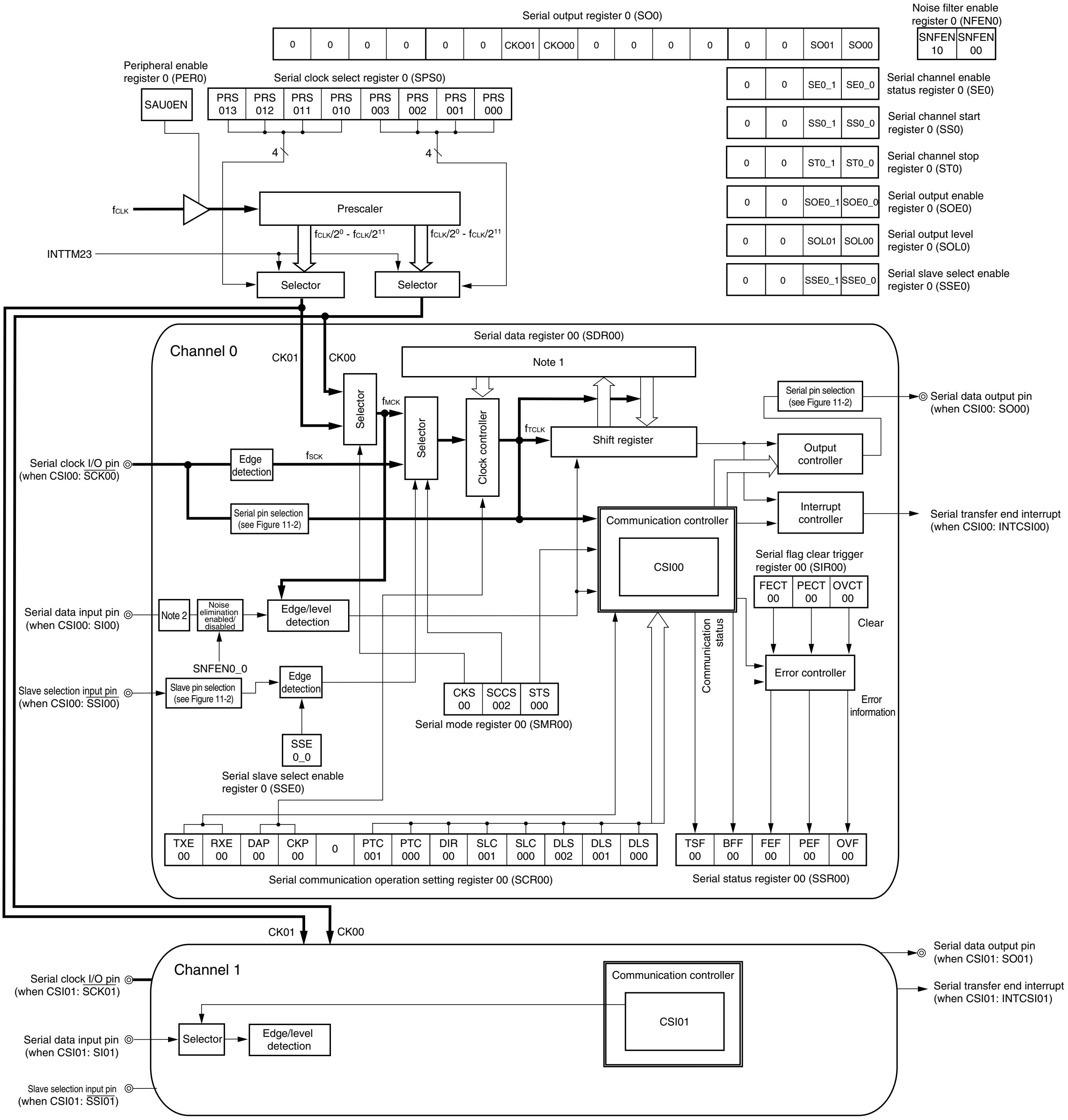
Item	Configuration
<R> Shift register	16 bits
Buffer register	Lower 8 bits of serial data register mn (SDRmn)[Note]
Serial clock I/O	$\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$, $\overline{\text{SCK11}}$ pins (for 3-wire serial I/O), SCL11, SCL20 pins (for simplified I^2C)
Serial data input	SI00, SI01, SI10, SI11, SI20, SI21 pins (for 3-wire serial I/O), RxD2 pins (for UART)
Serial data output	SO00, SO01, SO10, SO11 pins (for 3-wire serial I/O), TxD2 pins (for UART), output controller
Serial data I/O	SDA11, SDA20 pins (for simplified I^2C)
Slave select input	$\overline{\text{SSI00}}$, $\overline{\text{SSI01}}$ pins (for SPI)
Control registers	<Registers of unit setting block> • Peripheral enable register 0, 1 (PER0, PER1) • 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) • 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) • Serial slave select enable register 0 (SSE0) • Serial communication pin select register (STSEL) • Port input mode registers 6, 7 (PIM6, PIM7) • Port output mode registers 4, 7 (POM4, POM7) • Port mode registers 1, 3, 4, 6, 7, 15 (PM1, PM3, PM4, PM6, PM7, PM15) • Port registers 1, 3, 4, 6, 7, 15 (P1, P3, P4, P6, P7, P15)

Note During operation (SEm_n = 1)

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1),
p: CSI number (p = 00, 01, 10, 11), q: UART number (q = 2), r: IIC number (r = 11, 20)

Figure 11-1 shows the block diagram of serial array unit 0.

Figure 11-1. Block Diagram of Serial Array Unit 0


Serial output register 0 (SO0)
0 0 0 0 0 0 CKO01 CKO00 0 0 0 0 0 0 SO01 SO00
Noise filter enable register 0 (NFEN0)
SNFEN 10
SNFEN 00
Peripheral enable register 0 (PER0)
SAU0EN
Serial clock select register 0 (SPS0)
PRS 013
PRS 012
PRS 011
PRS 010
PRS 003
PRS 002
PRS 001
PRS 000
4
4
0 0 SE0_1 SE0_0
Serial channel enable status register 0 (SE0)
0 0 SS0_1 SS0_0
Serial channel start register 0 (SS0)
0 0 ST0_1 ST0_0
Serial channel stop register 0 (ST0)
0 0 SOE0_1 SOE0_0
Serial output enable register 0 (SOE0)
0 0 SOL01 SOL00
Serial output level register 0 (SOL0)
0 0 SSE0_1 SSE0_0
Serial slave select enable register 0 (SSE0)
fCLK
Prescaler
fCLK/2⁰ - fCLK/2¹¹
fCLK/2⁰ - fCLK/2¹¹
INTTM23
Selector
Selector
Channel 0
CK01
CK00
Serial data register 00 (SDR00)
Note 1
Selector
fMCK
Selector
Clock controller
fTCLK
Shift register
Serial pin selection (see Figure 11-2)
Serial data output pin (when CSI00: SO00)
Serial clock I/O pin (when CSI00: SCK00)
Edge detection
fSCK
Output controller
Serial pin selection (see Figure 11-2)
Communication controller
CSI00
Interrupt controller
Serial transfer end interrupt (when CSI00: INTCSI00)
Serial flag clear trigger register 00 (SIR00)
FECT 00
PECT 00
OVCT 00
Clear
Serial data input pin (when CSI00: SI00)
Note 2
Noise elimination enabled/ disabled
Edge/level detection
Communication status
SNFEN0_0
Slave selection input pin (when CSI00: SSI00)
Slave pin selection (see Figure 11-2)
Edge detection
CKS 00
SCCS 002
STS 000
Serial mode register 00 (SMR00)
Error controller
Error information
SSE 0_0
Serial slave select enable register 0 (SSE0)
TXE 00
RXE 00
DAP 00
CKP 00
0
PTC 001
PTC 000
DIR 00
SLC 001
SLC 000
DLS 002
DLS 001
DLS 000
Serial communication operation setting register 00 (SCR00)
TSF 00
BFF 00
FEF 00
PEF 00
OVF 00
Serial status register 00 (SSR00)
CK01
CK00
Channel 1
Serial data output pin (when CSI01: SO01)
Serial clock I/O pin (when CSI01: SCK01)
Communication controller
CSI01
Serial transfer end interrupt (when CSI01: INTCSI01)
Serial data input pin (when CSI01: SI01)
Selector
Edge/level detection
Slave selection input pin (when CSI01: SSI01)

Notes 1. When operation is stopped (SEm_n = 0), the higher 7 bits become the clock division setting section and the lower bits are fixed to 0.

During operation (SEm_n = 1), it becomes a buffer register.

2. Serial pin selection (see **Figure 11-2**)

<R>

Figure 11-2. Port Configuration Diagram of Serial Array Unit 0

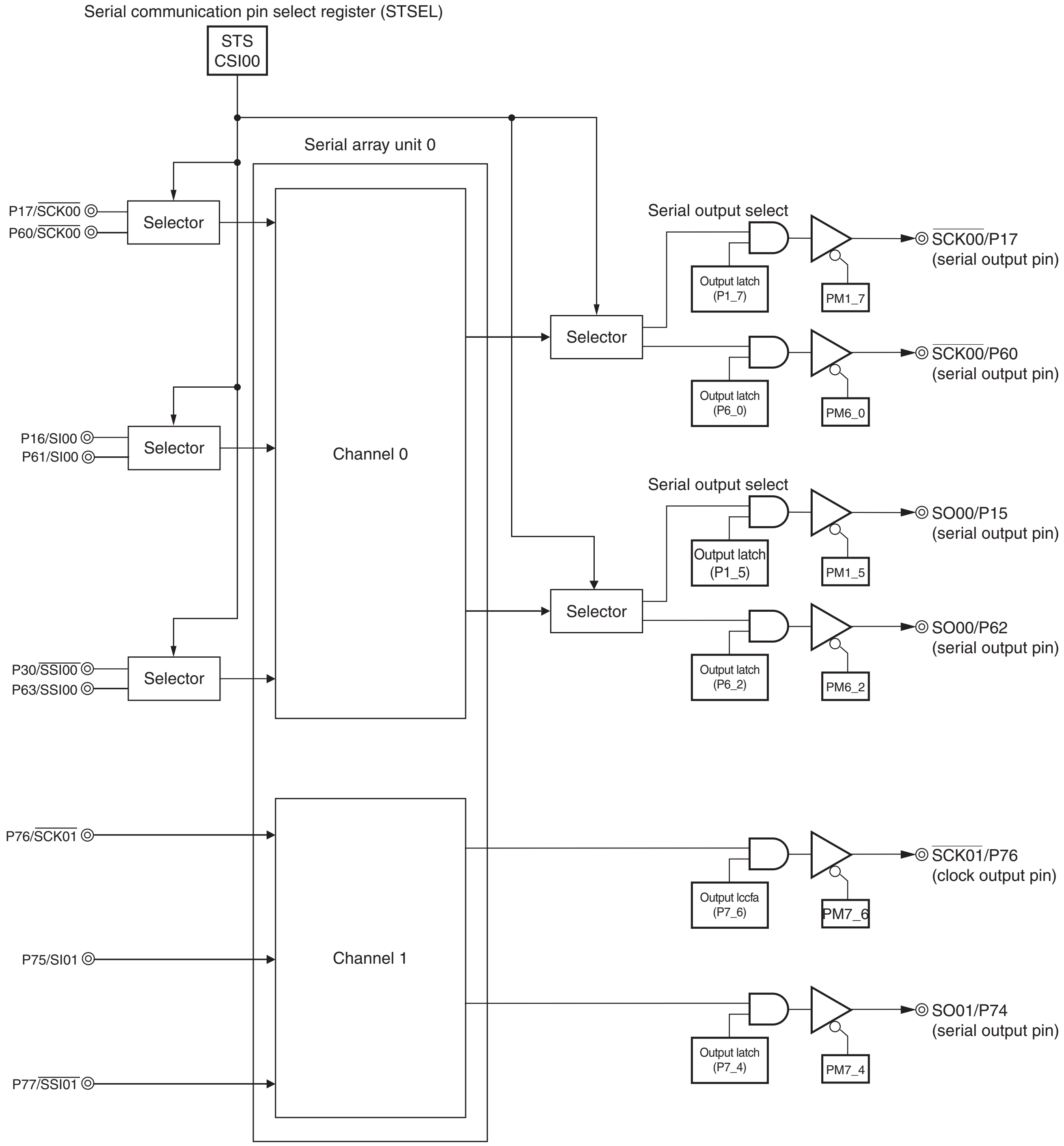
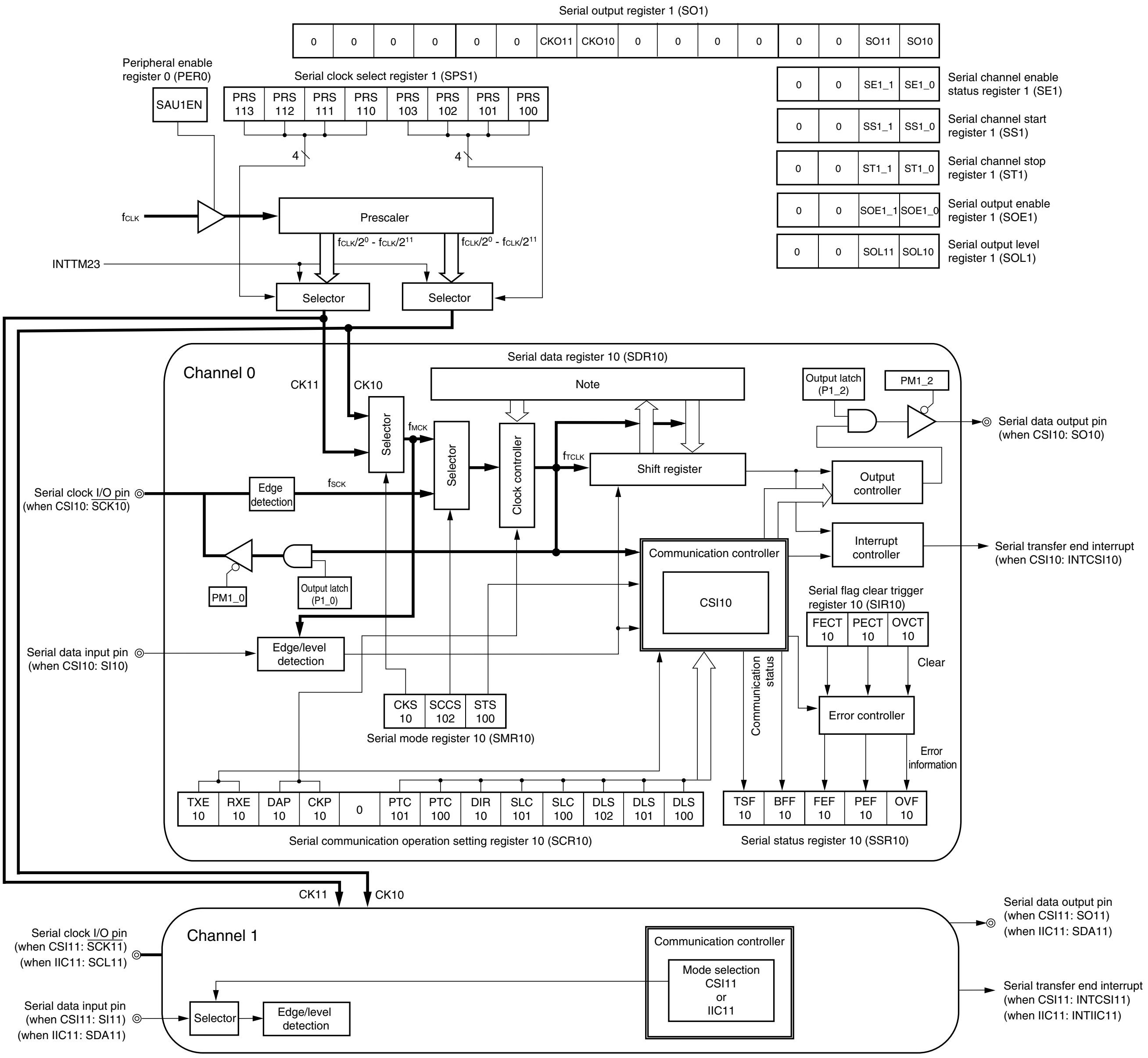

Serial communication pin select register (STSEL)
STS CSI00
Serial array unit 0
P17/$\overline{\text{SCK00}}$
P60/$\overline{\text{SCK00}}$
Selector
P16/SI00
P61/SI00
Selector
P30/$\overline{\text{SSI00}}$
P63/$\overline{\text{SSI00}}$
Selector
Channel 0
Selector
Selector
Serial output select
Output latch (P1_7)
PM1_7
$\overline{\text{SCK00}}$/P17 (serial output pin)
Output latch (P6_0)
PM6_0
$\overline{\text{SCK00}}$/P60 (serial output pin)
Serial output select
Output latch (P1_5)
PM1_5
SO00/P15 (serial output pin)
Output latch (P6_2)
PM6_2
SO00/P62 (serial output pin)
P76/$\overline{\text{SCK01}}$
P75/SI01
P77/$\overline{\text{SSI01}}$
Channel 1
Output lccfa (P7_6)
PM7_6
$\overline{\text{SCK01}}$/P76 (clock output pin)
Output latch (P7_4)
PM7_4
SO01/P74 (serial output pin)

Figure 11-3 shows the block diagram of serial array unit 1.

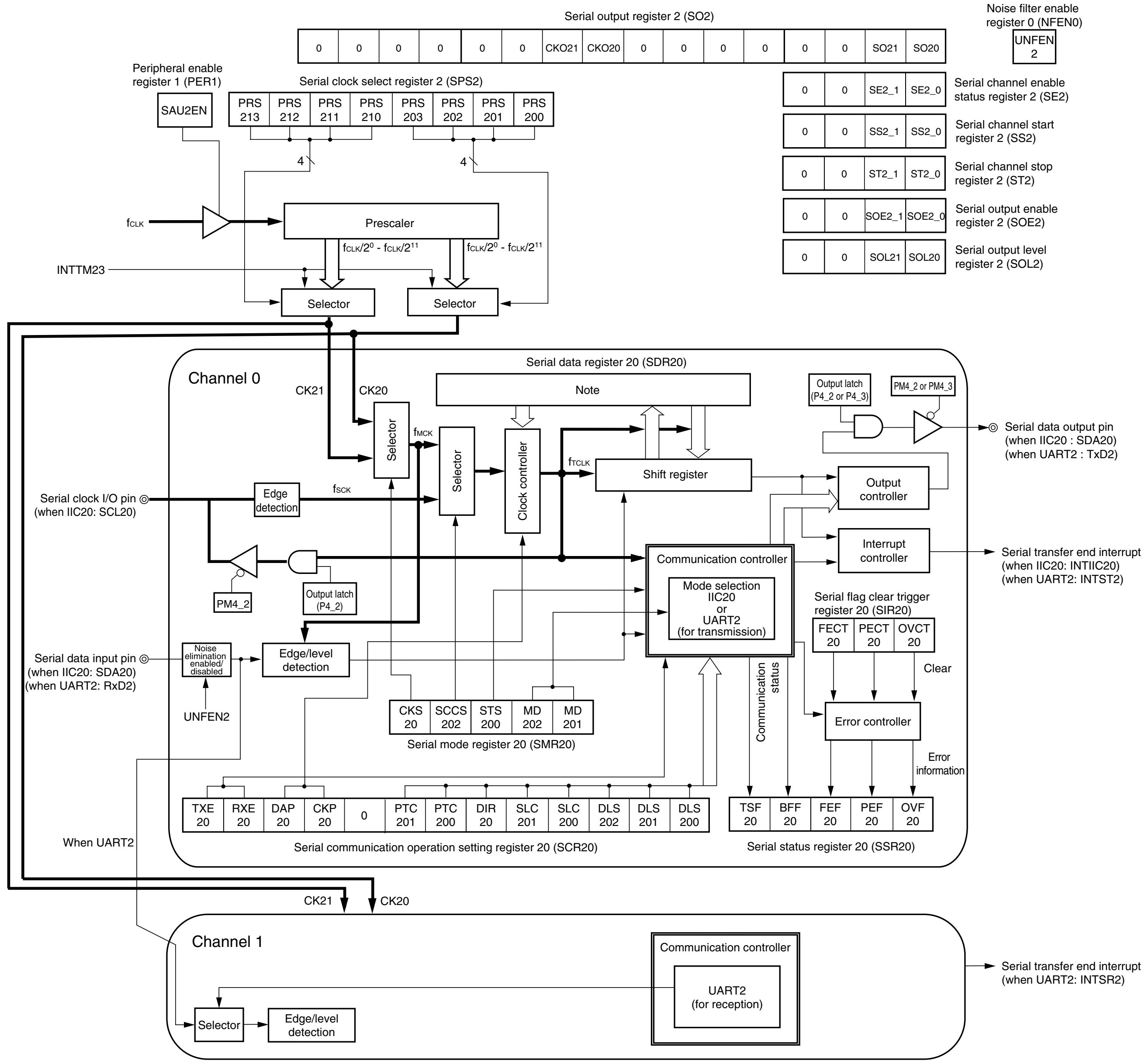
Figure 11-3. Block Diagram of Serial Array Unit 1


Serial output register 1 (SO1)
0 0 0 0 0 0 CKO11 CKO10 0 0 0 0 0 0 SO11 SO10
Peripheral enable register 0 (PER0)
SAU1EN
Serial clock select register 1 (SPS1)
PRS 113 PRS 112 PRS 111 PRS 110 PRS 103 PRS 102 PRS 101 PRS 100
4
4
0 0 SE1_1 SE1_0
Serial channel enable status register 1 (SE1)
0 0 SS1_1 SS1_0
Serial channel start register 1 (SS1)
0 0 ST1_1 ST1_0
Serial channel stop register 1 (ST1)
0 0 SOE1_1 SOE1_0
Serial output enable register 1 (SOE1)
0 0 SOL11 SOL10
Serial output level register 1 (SOL1)
fCLK
Prescaler
fCLK/2^0 - fCLK/2^11
fCLK/2^0 - fCLK/2^11
INTTM23
Selector
Selector
Channel 0
CK11
CK10
Serial data register 10 (SDR10)
Note
Output latch (P1_2)
PM1_2
Serial data output pin (when CSI10: SO10)
Selector
fMCK
Selector
Clock controller
fTCLK
Shift register
Output controller
Serial clock I/O pin (when CSI10: SCK10)
Edge detection
fSCK
Interrupt controller
Serial transfer end interrupt (when CSI10: INTCSI10)
Communication controller
CSI10
PM1_0
Output latch (P1_0)
Serial flag clear trigger register 10 (SIR10)
FECT 10 PECT 10 OVCT 10
Serial data input pin (when CSI10: SI10)
Edge/level detection
Clear
Communication status
Error controller
CKS 10 SCCS 102 STS 100
Serial mode register 10 (SMR10)
Error information
TXE 10 RXE 10 DAP 10 CKP 10 0 PTC 101 PTC 100 DIR 10 SLC 101 SLC 100 DLS 102 DLS 101 DLS 100
Serial communication operation setting register 10 (SCR10)
TSF 10 BFF 10 FEF 10 PEF 10 OVF 10
Serial status register 10 (SSR10)
CK11
CK10
Channel 1
Serial data output pin (when CSI11: SO11) (when IIC11: SDA11)
Serial clock I/O pin (when CSI11: SCK11) (when IIC11: SCL11)
Communication controller
Mode selection CSI11 or IIC11
Serial transfer end interrupt (when CSI11: INTCSI11) (when IIC11: INTIIC11)
Serial data input pin (when CSI11: SI11) (when IIC11: SDA11)
Selector
Edge/level detection

Note When operation is stopped (SEm_n = 0), the higher 7 bits become the clock division setting section and the lower bits are fixed to 0.
During operation (SEm_n = 1), it becomes a buffer register.

Figure 11-4 shows the block diagram of serial array unit 2.

Figure 11-4. Block Diagram of Serial Array Unit 2


Serial output register 2 (SO2)
0
0
0
0
0
0
CKO21
CKO20
0
0
0
0
0
0
SO21
SO20
Noise filter enable register 0 (NFEN0)
UNFEN 2
Peripheral enable register 1 (PER1)
SAU2EN
Serial clock select register 2 (SPS2)
PRS 213
PRS 212
PRS 211
PRS 210
PRS 203
PRS 202
PRS 201
PRS 200
4
4
0
0
SE2_1
SE2_0
Serial channel enable status register 2 (SE2)
0
0
SS2_1
SS2_0
Serial channel start register 2 (SS2)
0
0
ST2_1
ST2_0
Serial channel stop register 2 (ST2)
0
0
SOE2_1
SOE2_0
Serial output enable register 2 (SOE2)
0
0
SOL21
SOL20
Serial output level register 2 (SOL2)
fCLK
Prescaler
fCLK/2⁰ - fCLK/2¹¹
fCLK/2⁰ - fCLK/2¹¹
INTTM23
Selector
Selector
Channel 0
CK21
CK20
Serial data register 20 (SDR20)
Note
Output latch (P4_2 or P4_3)
PM4_2 or PM4_3
Serial data output pin (when IIC20 : SDA20) (when UART2 : TxD2)
Selector
fMCK
Selector
Clock controller
fTCLK
Shift register
Output controller
Serial clock I/O pin (when IIC20: SCL20)
Edge detection
fSCK
Interrupt controller
Serial transfer end interrupt (when IIC20: INTIIC20) (when UART2: INTST2)
Communication controller
Mode selection IIC20 or UART2 (for transmission)
PM4_2
Output latch (P4_2)
Serial flag clear trigger register 20 (SIR20)
FECT 20
PECT 20
OVCT 20
Clear
Serial data input pin (when IIC20: SDA20) (when UART2: RxD2)
Noise elimination enabled/ disabled
Edge/level detection
UNFEN2
CKS 20
SCCS 202
STS 200
MD 202
MD 201
Serial mode register 20 (SMR20)
Communication status
Error controller
Error information
TXE 20
RXE 20
DAP 20
CKP 20
0
PTC 201
PTC 200
DIR 20
SLC 201
SLC 200
DLS 202
DLS 201
DLS 200
Serial communication operation setting register 20 (SCR20)
TSF 20
BFF 20
FEF 20
PEF 20
OVF 20
Serial status register 20 (SSR20)
When UART2
CK21
CK20
Channel 1
Communication controller
UART2 (for reception)
Serial transfer end interrupt (when UART2: INTSR2)
Selector
Edge/level detection

Note When operation is stopped (SEm_n = 0), the higher 7 bits become the clock division setting section and the lower bits are fixed to 0.
During operation (SEm_n = 1), it becomes a buffer register.

(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, serial data register mn (SDRmn) is used during operation (SEm_n = 1).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Shift register																

(2) Serial data register mn (SDRmn)

SDRmn is the transmit/receive data register (16 bits) of channel n. When operation is stopped (SEm_n = 0), bits 15 to 9 are used as the division setting register of the operating clock (f_{MCK}). During operation (SEm_n = 1), bits 15 to 9 are used as a transmission/reception buffer register.
When data is received, parallel data converted by the shift register is stored. When data is to be transmitted, set transmit to be transferred to the shift register.
The data stored in this register is as follows, depending on the setting of bits 3 to 0 (DLSmn3 to DLSmn0) of the SCRmn register, regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of SDRmn register) (settable in UART mode only)
- 8-bit data length (stored in bits 0 to 7 of SDRmn register)

:

- 16-bit data length (stored in bits 0 to15 of SDRmn register)

SDRmn can be read or written in 16-bit units.
When SEm_n = 1, the lower 8 bits of SDRmn can be read or written[Note] in 8-bit units as SDRmnL. The SDRmnL registers that can be used according to the communication methods are shown below.

- CSIp communication … SDRpL
- UARTq reception … SDR21L
- UARTq transmission … SDR20L (UARTq transmit data register)
- IICr communication … SDRrL (IICr data register)

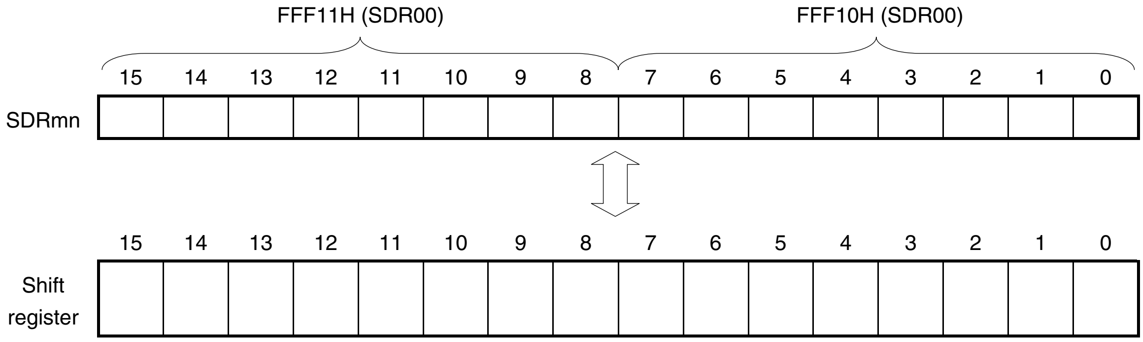
Reset signal generation clears this register to 0000H.

Note Writing in 8-bit units is prohibited when the operation is stopped (SEm_n = 0).

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1),
p: CSI number (p = 00, 01, 10, 11), q: UART number (q = 2), r: IIC number (r = 11, 20)

Figure 11-5. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), After reset: 0000H R/W
FFF14H, FFF45H (SDR10), FFF16H, FFF17H (SDR11),
FFF48H, FFF49H (SDR20), FFF46H, FFF47H (SDR21)



Remarks 1. For the function of the higher 7 bits of SDRmn, see **11.3 Registers Controlling Serial Array Unit**.

2. m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1),
p: CSI number (p = 00, 01, 10, 11), q: UART number (q = 2), r: IIC number (r = 11, 20)

11.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0, 1 (PER0, PER1)
- 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)
- Serial slave select enable register 0 (SSE0)
- Serial communication pin select register (STSEL)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 6, 7 (PIM6, PIM7)
- Port output mode registers 4, 7 (POM4, POM7)
- Port mode registers 1, 3, 4, 6, 7, 15 (PM1, PM3, PM4, PM6, PM7, PM15)
- Port registers 1, 3, 4, 6, 7, 15 (P1, P3, P4, P6, P7, P15)

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

(1) Peripheral enable registers 0, 1 (PER0, PER1)

PER0, PER1 are 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 3 (SAU0EN) of PER0 to 1.
When serial array unit 1 is used, be sure to set bit 4 (SAU1EN) of PER0 to 1.
When serial array unit 2 is used, be sure to set bit 3 (SAU2EN) of PER1 to 1.
PER0, PER1 can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 11-6. Format of Peripheral Enable Registers 0, 1 (PER0, PER1)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	ADCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN

Address: F00F1H After reset: 00H R/W

Symbol	7	6	5	4	<3>	2	<1>	<0>
PER1	0	0	0	0	SAU2EN	0	WUTEN	DELEN

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

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.

SAU2EN	Control of serial array unit 2 input clock
0	Stops supply of input clock. • SFR used by serial array unit 2 cannot be written. • Serial array unit 2 is in the reset status.
1	Supplies input clock. • SFR used by serial array unit 2 can be read/written.

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

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 noise filter enable register (NFEN0), Serial communication pin select register (STSEL), port input mode register (PIM6, PIM7), port output mode register (POM4, POM7), port mode registers (PM1, PM3, PM4, PM6, PM7, PM15), and port registers (P1, P3, P4, P6, P7, P15)).

2. After setting the PER0, PER1 registers to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

3. Be sure to clear the following bits to 0.
78K0R/HC3: Bit 2 of the PER0 register, bits 2 to 7 of the PER1 register
78K0R/HE3, 78K0R/HF3, 78K0R/HG3: Bits 2, 4 to 7 of the PER1 register

Remark m: Unit number (m = 0 to 2)

(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 SEm_n = 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 11-7. Format of Serial Clock Select Register m (SPSm)

Address: F0116H, F0117H (SPS0), F0146H, F0147H (SPS1) After reset: 0000H R/W
F0176H, F0177H (SPS2)

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]				
					f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	4 MHz	8 MHz	16 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	2 MHz	4 MHz	8 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	1 MHz	2 MHz	4 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	500 kHz	1 MHz	2 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	250 kHz	500 kHz	1 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	125 kHz	250 kHz	500 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	62.5 kHz	125 kHz	250 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	31.3 kHz	62.5 kHz	125 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	15.6 kHz	31.3 kHz	62.5 kHz	93.75 kHz
1	0	0	1	$f_{CLK}/2^9$	7.81 kHz	15.6 kHz	31.3 kHz	46.88 kHz
1	0	1	0	$f_{CLK}/2^{10}$	3.91 kHz	7.81 kHz	15.6 kHz	23.44 kHz
1	0	1	1	$f_{CLK}/2^{11}$	1.95 kHz	3.91 kHz	7.81 kHz	11.72 kHz
1	1	1	1	INTTM23				
Other than above				Setting prohibited				

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (STm = 0003H) the operation of the serial array unit m (SAUm). When selecting INTTM23 for the operation clock, also stop the timer array unit 2 (TAU2) (TT2 = 00FFH).

Cautions 1. Be sure to clear bits 15 to 8 to "0".

2. After setting the PER0, PER1 registers 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

2. m: Unit number (m = 0 to 2), 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 (f_{MCK}), specify whether the serial clock (f_{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 SEm_n = 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 11-8. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0108H, F0109H (SMR00), F010AH, F010BH (SMR01), After reset: 0020H R/W
F0138H, F0139H (SMR10), F013AH, F013BH (SMR11),
F0168H, F0169H (SMR20), F016AH, F016BH (SMR21)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	SCCS mn	0	0	0	0	0	STS mn	0	SIS mn0	1	0	0	MD mn2	MD mn1	MD mn0

CKS mn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by SPSm register
1	Operation clock CKm1 set by SPSm register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

SCCS mn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided clock of operation clock f_{MCK} specified by CKSmn bit
1	Clock input from $\overline{SCK}$ pin (slave transfer in CSI mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of f_{MCK} is set by the higher 7 bits of the SDRmn register.	

STS mn	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I^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 to 2), n: Channel number (n = 0, 1)

Figure 11-8. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0108H, F0109H (SMR00), F010AH, F010BH (SMR01), After reset: 0020H R/W
F0138H, F0139H (SMR10), F013AH, F013BH (SMR11),
F0168H, F0169H (SMR20), F016AH, F016BH (SMR21)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	SCCS 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[Note]
0	0	CSI mode
0	1	UART mode
1	0	Simplified I^{2}C mode
1	1	Setting prohibited

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting MDmn0 to 1 when SDRmn data has run out.	

Note See **Table 11-1 Serial Function Assignment of Each Product** for details of the modes implemented for each unit and product.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

(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 SEm_n = 1).

SCRmn can be set by a 16-bit memory manipulation instruction.

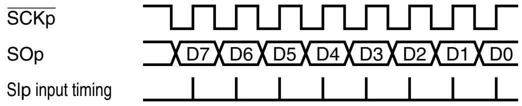
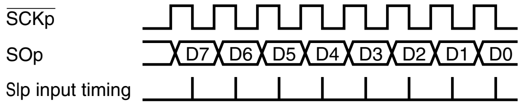
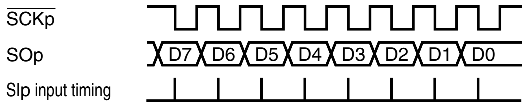
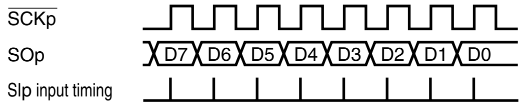
Reset signal generation sets this register to 0087H.

Figure 11-9. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/3)

Address: F010CH, F010DH (SCR00), F010EH, F010FH (SCR01), After reset: 0087H R/W
F013CH, F013DH (SCR10), F013EH, F013FH (SCR11),
F016CH, F016DH (SCR20), F016EH, F016FH (SCR21)

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	0	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	DLS mn3	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	 $\overline{\text{SCKp}}$ SOp SIp input timing D7 D6 D5 D4 D3 D2 D1 D0	1
0	1	 $\overline{\text{SCKp}}$ SOp SIp input timing D7 D6 D5 D4 D3 D2 D1 D0	2
1	0	 $\overline{\text{SCKp}}$ SOp SIp input timing D7 D6 D5 D4 D3 D2 D1 D0	3
1	1	 $\overline{\text{SCKp}}$ SOp SIp input timing D7 D6 D5 D4 D3 D2 D1 D0	4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C mode.			

Note Be sure to clear this bit to 0 for serial array units 0 and 1.

Caution Be sure to clear bits 6, 10, and 11 to “0”.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 11-9. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/3)

Address: F010CH, F010DH (SCR00), F010EH, F010FH (SCR01), After reset: 0087H R/W
F013CH, F013DH (SCR10), F013EH, F013FH (SCR11),
F016CH, F016DH (SCR20), F016EH, F016FH (SCR21)

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	0	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	DLS mn3	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[Note].	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^{2}C mode.			

DIR mn	Selection of data transfer sequence in CSI and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIRmn = 0 in the simplified I^{2}C mode.	

Note 0 is always added regardless of the data contents.

Caution Be sure to clear bits 6, 10, and 11 to “0”.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

Figure 11-9. Format of Serial Communication Operation Setting Register mn (SCRmn) (3/3)

Address: F010CH, F010DH (SCR00), F010EH, F010FH (SCR01), After reset: 0087H R/W
F013CH, F013DH (SCR10), F013EH, F013FH (SCR11),
F016CH, F016DH (SCR20), F016EH, F016FH (SCR21)

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	0	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	DLS mn3	DLS mn2	DLS mn1	DLS mn0

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^{2}C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the CSI mode.		

DLS mn3	DLS mn2	DLS mn1	DLS mn0	Setting of data length in CSI and UART modes	Serial-function correspondence		
					CSI	UART	IIC
0	1	1	0	7-bit data length (stored in bits 0 to 6 of SDRmn register)	√	√	–
0	1	1	1	8-bit data length (stored in bits 0 to 7 of SDRmn register)	√	√	√
1	0	0	0	9-bit data length (stored in bits 0 to 8 of SDRmn register)	√	√	–
1	0	0	1	10-bit data length (stored in bits 0 to 9 of SDRmn register)	√	–	–
1	0	1	0	11-bit data length (stored in bits 0 to 10 of SDRmn register)	√	–	–
1	0	1	1	12-bit data length (stored in bits 0 to 11 of SDRmn register)	√	–	–
1	1	0	0	13-bit data length (stored in bits 0 to 12 of SDRmn register)	√	–	–
1	1	0	1	14-bit data length (stored in bits 0 to 13 of SDRmn register)	√	–	–
1	1	1	0	15-bit data length (stored in bits 0 to 14 of SDRmn register)	√	–	–
1	1	1	1	16-bit data length (stored in bits 0 to 15 of SDRmn register)	√	√	–
Other than above				Setting prohibited			
Be sure to set DLSmn3 to DLSmn0 = 0111B in the simplified I^{2}C mode.							

Caution Be sure to clear bits 6, 10, and 11 to “0”.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

(5) Higher 7 bits of the serial data register mn (SDRmn)

SDRmn is the transmit/receive data register (16 bits) of channel n. When operation is stopped (SEm_n = 0), bits 15 to 9 are used as the division setting register of the operating clock (f_{MCK}). During operation (SEm_n = 1), bits 15 to 9 are used as a transmission/reception buffer register.

If the SCCSmn 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.

See **11.2 Configuration of Serial Array Unit** for the functions of SDRmn during operation (SEm_n = 1).

SDRmn can be read or written in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 11-10. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), After reset: 0000H R/W
FFF14H, FFF15H (SDR10), FFF16H, FFF17H (SDR11),
FFF44H, FFF45H (SDR20), FFF46H, FFF47H (SDR21)

	FFF11H (SDR00)								FFF10H (SDR00)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0	0	0	0	0	0	0	0	0

SDRmn[15:9]							Setting of division ratio of operation clock (f_{MCK})
0	0	0	0	0	0	0	f_{MCK}/2
0	0	0	0	0	0	1	f_{MCK}/4
0	0	0	0	0	1	0	f_{MCK}/6
0	0	0	0	0	1	1	f_{MCK}/8
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	f_{MCK}/254
1	1	1	1	1	1	1	f_{MCK}/256

Cautions 1. When operation is stopped (SEm_n = 0), be sure to clear bits 8 to 0 to “0”.

2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

3. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I^2C is used. Set SDRmn[15:9] to 0000001B or greater.

4. Do not write eight bits to the lower eight bits if operation is stopped (SEm_n = 0). (If these bits are written to, the higher seven bits are cleared to 0.)

Remarks 1. For the function of during operation (SEm_n = 1), see **11.2 Configuration of Serial Array Unit**.

2. m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

(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 11-11. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01),   After reset: 0000H   R
F0130H, F0131H (SSR10), F0132H, F0133H (SSR11),
F0160H, F0161H (SSR20), F0162H, F0163H (SSR21

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	PEF mn	OVF mn

TSF mn	Communication status indication flag of channel n
0	• When the STm_n and SSm_n bits are set to “1” • 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 STm_n/SSm_n bit is set to 1.	

BFF mn	Buffer register status indication flag of channel n
0	• When the STm_n and SSm_n bits are set to “1” • 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 STm_n/SSm_n 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.	

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

Figure 11-11. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01), After reset: 0000H R
F0130H, F0131H (SSR10), F0132H, F0133H (SSR11),
F0160H, F0161H (SSR20), F0162H, F0163H (SSR21

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	PEF mn	OVF mn

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^{2}C 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^{2}C 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.	

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

(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 11-12. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0104H, F0105H (SIR00), F0106H, F0107H (SIR01), After reset: 0000H R/W
F0134H, F0135H (SIR10), F0136H, F0137H (SIR11),
F0164H, F0165H (SIR20), F0166H, F0167H (SIR21)

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.

Cautions 1. Be sure to clear bits 15 to 3 to “0”.

2. Only the error flag set to the SSRn register is cleared by using the SIRmn register. When a clear operation is performed for an error flag that is not set and when a new error is detected between reading the error flag and the clear operation, the error flag may be erased.

Remarks 1. When the SIRmn register is read, 0000H is always read.

2. When writing “1” to a clear trigger and setting (1) the corresponding error flag occur simultaneously, setting the error flag takes precedence.

3. m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

(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 m (SSm), the corresponding bit of this register is set to 1. When 1 is written a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of 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.

<R> **Figure 11-13. Format of Serial Channel Enable Status Register m (SEm)**

Address: F0110H, F0111H (SE0), F0140H, F0141H (SE1) After reset: 0000H R
F0170H, F0171H (SE2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE m_ 1	SE m_ 0

SE m_n	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.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

(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 (SSm_n), the corresponding bit (SEm_n) of serial channel enable status register m (SEm) is set to 1. Because SSm_n is a trigger bit, it is cleared immediately when SEm_n = 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.

<R> **Figure 11-14. Format of Serial Channel Start Register m (SSm)**

Address: F0112H, F0113H (SS0), F0132H, F0133H (SS1), After reset: 0000H R/W
F0172H, F0173H (SS2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS m_1	SS m_0

SS m_n	Operation start trigger of channel n
0	No trigger operation
1	Sets SEm_n 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 2 to "0".

Remarks 1. When the SSm register is read, 0000H is always read.

2. m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

(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 (STm_n), the corresponding bit (SEm_n) of serial channel enable status register m (SEm) is cleared to 0. Because STm_n is a trigger bit, it is cleared immediately when SEm_n = 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.

<R> **Figure 11-15. Format of Serial Channel Stop Register m (STm)**

Address: F0114H, F0115H (ST0), F0144H, F0145H (ST1), After reset: 0000H R/W
F0174H, F0175H (ST2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST m_ 1	ST m_ 0

ST m_n	Operation stop trigger of channel n
0	No trigger operation
1	Clears SEm_n 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 2 to “0”.

Remarks 1. When the STm register is read, 0000H is always read.

2. m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

(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.

<R> **Figure 11-16. Format of Serial Output Enable Register m (SOEm)**

Address: F011AH, F011BH (SOE0), F014AH, F014BH (SOE1), After reset: 0000H R/W
F017AH, F017BH (SOE2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE m_1	SOE m_0

SOE m_n	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 2 of SOEm.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1),
mn = 00, 01, 10, 11, 20, 21

(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 (SOEm_n = 0). When serial output is enabled (SOEm_n = 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 (SEm_n = 0). While channel operation is enabled (SEm_n = 1), rewriting by software is ignored, and the value of CKOmn can be changed only by a serial communication operation.

To use the P10/INTP4/SCK10/CTxD/LTxD1/TI00/TO00, P12/SO10/INTP3/TI16/TO16, P15/SO10/TI10/TO10, P17/SCK00/TI14/TO14, P42/TxD2/SCL20, P60/SCK00/SCL11, P62/SO00, P74/SO01/KR4, P76/SCK01/KR6, P151/SO11, or P153/SCK11 pin as a port function pin, set the corresponding CKOmn and SOmn bits to "1" regardless of the serial communication state, because a low level is output when these bits are set to "0".

SOm can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0303H.

Figure 11-17. Format of Serial Output Register m (SOm)

Address: F0118H, F0119H (SO0), F0148H, F0149H (SO1), After reset: 0303H R/W
F0178H, F0179H (SO2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKO m1	CKO m0	0	0	0	0	0	0	SO m1	SO m0

CKO mn	Serial clock output of channel n
0	Serial clock output value is "0".
1	Serial clock output value is "1".

SO mn	Serial data output of channel n
0	Serial data output value is "0".
1	Serial data output value is "1".

Caution Be sure to set bits 15 to 10, 7 to 2 of SOm to "1".

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1),
mn = 00, 01, 10, 11, 20, 21

(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 (SOEm_n = 1). When serial output is disabled (SOEm_n = 0), the value of the SOmn bit is output as is.

Rewriting SOLm is prohibited when the register is in operation (when SEm_n = 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 11-18. Format of Serial Output Level Register m (SOLm)

Address: F0120H, F0121H (SOL0), F0150H, F0151H (SOL1), After reset: 0000H R/W
F0158H, F0159H (SOL2)

Symbol	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	0	SOL m1	SOL m0

SOL mn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution Be sure to clear bits 15 to 2 to "0".

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

(14) Serial slave select enable register 0 (SSE0)

The SSE0 register controls the SSI pin input of a channel during CSI communication and in slave mode.

While a high level is being input to the SSI pin, no transmission/reception operation is performed even if a serial clock is input. While a low level is being input to the SSI pin, a transmission/reception operation is performed according to each mode setting if a serial clock is input.

Reset signal generation clears this register to 0000H.

Cautions 1. Writing is prohibited except during CSI communication and in slave mode.

2. Settable only when SAU is stopped (SE0 = 0).

3. Only serial array unit 0 (SAU0)

<R> **Figure 11-19. Format of Serial Slave Select Enable Register (SSE0)**

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
SSE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSE 0_1	SSE 0_0

SSE0_n	Channel n $\overline{\text{SSI}}$ input setting in CSI communication and slave mode
0	Disables $\overline{\text{SSI0n}}$ pin input.
1	Enables $\overline{\text{SSI0n}}$ pin input.

Caution Be sure to clear bits 15 to 2 to “0”.

Remark n = 0, 1

(15) Serial communication pin select register (STSEL)

The STSEL register is used to switch the input source to the timer array unit and the serial array unit and UART2 communication pins.

The CSI00 communication pin can be selected by using bit 4.

The IIC11 communication pin can be selected by using bit 5.

This register can be read or written in 1-bit units or 8-bit units.

Figure 11-20. Serial communication pin select register (STSEL)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
STSEL	STSLIN1	TMCAN	STSIIC11[Note]	STSCSI00	TM30K	TMLIN1	TMLIN0	0

STSIIC11[Note]	IIC11 communication pin selection	
	SCL11	SDA11
1	P60	P61

STSCSI00	CSI00 communication pin selection			
	$\overline{\text{SSI00}}$	$\overline{\text{SCK00}}$	SI00	SO00
0	P30	P17	P16	P15
1	P63	P60	P61	P62

Note When using IIC11, be sure to set STSIIC11 to “1”. In addition, if not using the alternate functions, clear STSIIC11 to 0.

(16) 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.

When using the 78K0R/Hx3 as a slave during CSI communication, set the bits corresponding to the $\overline{\text{SSI00}}$ and $\overline{\text{SSI01}}$ pins to 1 to enable the noise filters for these pins.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

<R> When the noise filter is enabled, Operating clock (f_{MCK}) 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 11-21. 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	0	0	UNFEN2	0	0	SNFEN01	SNFEN00

UNFEN2	Use of noise filter of RxD2/P43 pin
0	Noise filter OFF
1	Noise filter ON
Set UNFEN2 to 1 to use the RxD2 pin. Clear UNFEN2 to 0 to use the P43 pin.	

SNFEN01	Use of noise filter of $\overline{\text{SSI01}}$/P77 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN01 to 1 to use the $\overline{\text{SSI01}}$ pin. Clear SNFEN01 to 0 to use the P77 pin.	

SNFEN00	Use of noise filter of $\overline{\text{SSI00}}$/P30 or $\overline{\text{SSI00}}$/P63 pins[Note]
0	Noise filter OFF
1	Noise filter ON
Set SNFEN00 to 1 to use the $\overline{\text{SSI00}}$ pin. Clear SNFEN00 to 0 to use the P30, and P63 pins.	

Note Selected by using the STSCSI00 bit.

Caution Be sure to clear bits 7 to 5, 3, and 2 to "0".

(17) Port input mode registers 6, 7 (PIM6, PIM7)

These registers set the input buffer of ports 6, and 7 in 1-bit units.

PIM6, and PIM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

<R> **Figure 11-22. Format of Port Input Mode Registers 6, and 7 (PIM6, PIM7)**

Address F0046H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM6	0	0	0	0	PIM6_3	0	PIM6_1	PIM6_0

Address F0047H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM7	PIM7_7	PIM7_6	PIM7_5	0	PIM7_3	0	0	0

PIMm_n	Pmn pin input buffer selection (m = 6, 7; n = 0, 1, 3, 5 to 7)
0	Normal input buffer
1	TTL input buffer

(18) Port output mode registers 4, 7 (POM4, POM7)

These registers set the output mode of ports 4, and 7 in 1-bit units.

POM4, and POM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

<R> **Figure 11-23. Format of Port Output Mode Registers 4, and 7 (POM4, POM7)**

Address F0054H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM4	0	0	0	0	POM4_3	POM4_2	0	0

Address F0057H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM7	0	POM7_6	0	POM7_4	0	POM7_2	0	0

POMm_n	Pmn pin output buffer selection (m = 4, 7; n = 2 to 4, 6)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

(19) Port mode registers 1, 3, 4, 6, 7, 15 (PM1, PM3, PM4, PM6, PM7, PM15)

These registers set input/output of ports 1, 3, 4, 6, 7, and 15 in 1-bit units.

When using the pins for serial data/serial clock output or serial data/serial clock input, set the port registers and port mode registers as shown in Table 11-3.

PM1, PM3, PM4, PM6, PM7, and PM15 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Table 11-3. Port Register and Port Mode Register Settings When Using Serial

Pin Name	Pin Setting	Port Register	Port Mode Register
P10/SCK10/CTxD/LTxD1/TI00/TO00	Serial clock input	×	1
	Serial clock output	1	0
P11/SI10/CRxD/LRxD1/INTPLR1/TI02/TO02	Serial data input	×	1
P12/SO10/INTP3/TI16/TO16	Serial data output	1	0
P15/SO00/TI10/TO10	Serial data output	1	0
P16/SI00/TI12/TO12	Serial data input	×	1
P17/SCK00/TI14/TO14	Serial clock output	×	1
	Serial clock input	1	0
P30/SSI00/INTP2/TI01/TO01	Serial data input	×	1
P42/TxD2/SCL20	Serial clock output	1	0
P43/RxD2/INTPR2/SDA20	Serial data input	×	1
	Serial data output	1	0
P60/SCK00/SCL11	Serial clock input	×	1
	Serial clock output	1	0
P61/SI00/SDA11	Serial data input	×	1
	Serial data output	1	0
P62/SO00	Serial data output	1	0
P63/SSI00	Serial data input	×	1
P74/SO01/KR4	Serial data output	1	0
P75/SI01/KR5	Serial data input	×	1
P76/SCK01/KR6	Serial clock input	×	1
	Serial clock output	1	0
P77/SSI01/KR7	Serial data input	×	1
P151/SO11	Serial data output	1	0
P152/SI11	Serial data input	×	1
P153/SCK11	Serial clock input	×	1
	Serial clock output	1	0

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark X: Don't care

<R>

Figure 11-24. Format of Port Mode Registers 1, 3, 4, 6, 7, and 15 (PM1, PM3, PM4, PM6, PM7, PM15)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM1_7	PM1_6	PM1_5	PM1_4	PM1_3	PM1_2	PM1_1	PM1_0

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	PM3_2	PM3_1	PM3_0

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	PM4_7	PM4_6	PM4_5	PM4_4	PM4_3	PM4_2	PM4_1	PM4_0

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	PM6_7	PM6_6	PM6_5	PM6_4	PM6_3	PM6_2	PM6_1	PM6_0

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM7_7	PM7_6	PM7_5	PM7_4	PM7_3	PM7_2	PM7_1	PM7_0

Address: FFF2FH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM15	PM15_7	PM15_6	PM15_5	PM15_4	PM15_3	PM15_2	PM15_1	PM15_0

PMm_n	Pmn pin I/O mode selection (m = 1, 3, 4, 6, 7, 15; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

11.4 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/SCK10/CTxD/LTxD1/TI00/TO00, P11/SI10/CRxD/LRxD1/INTPLR1/TI02/TO02, P12/SO10/INTP2/TI16/TO16, P15/SO00/TI10/TO10, P16/SI00/TI12/TO12, P17/SCK00/TI14/TO14, P30/SSI00/INTP2/TI01/TO01, P42/TxD2/SCL20, P43/RxD2/INTPR2/SDA20, P60/SCK00/SCL11, P61/SI00/SDA11, P62/SO00, P63/SSI00, P74/SO01/KR4, P75/SI01/KR5, P76/SCK01/KR6, P77/SSI01/KR7, P151/SO11, P152/SI11, or P153/SCK11 pin can be used as ordinary port pins in this mode.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

11.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable registers 0, 1 (PER0, PER1).

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 3 (SAU0EN) of PER0 to 0.

To stop the operation of serial array unit 1, set bit 4 (SAU1EN) of PER0 to 0.

To stop the operation of serial array unit 2, set bit 3 (SAU2EN) of PER1 to 0.

Figure 11-25. Peripheral Enable Registers 0, 1 (PER0, PER1) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAU0, SAU1 to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	ADCEN ×	LIN1EN ×	LIN0EN ×	SAU1EN 0/1	SAU0EN 0/1	TAU2EN ×	TAU1EN ×	TAU0EN ×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

(b) Peripheral enable register 1 (PER1) … Set only the bit of SAU2 to be stopped to 0.

	7	6	5	4	3	2	1	0
PER1	0	0	0	0	SAU2EN 0/1	0	WUTEN ×	DFLEN ×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

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

Cautions 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read (except for noise filter enable register (NFEN0), Serial communication pin select register (STSEL), port input mode register (PIM6, PIM7), port output mode register (POM4, POM7), port mode registers (PM1, PM3, PM4, PM6, PM7, PM15), and port registers (P1, P3, P4, P6, P7, P15)).

2. Be sure to clear the following bits to 0.
78K0R/HC3: Bit 2 of the PER0 register, bits 2 to 7 of the PER1 register
78K0R/HE3, 78K0R/HF3, 78K0R/HG3: Bits 2, 4 to 7 of the PER1 register

Remark m: Unit number (m = 0, 1)
×: 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

11.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 11-26. Each Register Setting When Stopping the Operation by Channels

(a) Serial Channel Enable Status Register m (SEm) … This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SEm_1 0/1	SEm_0 0/1

0: Operation stops

* The SEm register is a read-only status register, whose operation is stopped by using the STm register.
With a channel whose operation is stopped, the value of CKOmn of the SOm 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.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	STm_1 0/1	STm_0 0/1

1: Clears SEm_n to 0 and stops the communication operation

* Because STm_n is a trigger bit, it is cleared immediately when SEm_n = 0.

(c) Serial output enable register m (SOEm) … This register is a register that is used to enable or stop output of the serial communication operation of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm_1 0/1	SOEm_0 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.

(d) Serial output register m (SOm) …This register is a buffer register for serial output of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

1: Serial clock output value is “1”

1: Serial data output value is “1”

* When using pins corresponding to each channel as port function pins, set the corresponding CKO0n and SO0n bits to “1”.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)
▨ : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-26. Each Register Setting When Stopping the Operation by Channels (2/2)

(e) Serial slave select enable register 0 (SSE0) ··· Control of the SSI pin of each slave channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSE0_1 0/1	SSE0_0 0/1

0: Disables the input value of the $\overline{\text{SSI}}$ pin

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

▨ : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

11.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) Communication

This is a clocked communication function that uses three lines: serial clock ($\overline{\text{SCK}}$) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 to 16 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

The channels supporting 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11) are channels 0 and 1 of SAU0 and channels 0 and 1 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^{2}C
0	0	CSI00 (supports SPI)	–	–
	1	CSI01 (supports SPI)		–
1	0	CSI10	–	–
	1	CSI11		IIC11
2	0	–	UART2	IIC20
	1	–		–

3-wire serial I/O (CSI00, CSI01, CIS10, CIS11) performs the following six types of communication operations.

- Master transmission (See **11.5.1**.)
- Master reception (See **11.5.2**.)
- Master transmission/reception (See **11.5.3**.)
- Slave transmission (See **11.5.4**.)
- Slave reception (See **11.5.5**.)
- Slave transmission/reception (See **11.5.6**.)

11.5.1 Master transmission

Master transmission is that the 78K0R/Hx3 outputs a transfer clock and transmits data to another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	$\overline{\text{SCK00}}$, SO00	$\overline{\text{SCK01}}$, SO01	$\overline{\text{SCK10}}$, SO10	$\overline{\text{SCK11}}$, SO11
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11
	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 to 16 bits			
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz][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 29 ELECTRICAL SPECIFICATIONS**).

(1) Register setting

Figure 11-27. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

Communication starts when these bits are 1 if the data phase is forward (CKPmn = 0). If the phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(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	0	SOEm_1 0/1	SOEm_0 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	0	0	SSm_1 0/1	SSm_0 0/1

(d) 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	SCCSmn 0	0	0	0	0	0	STSmn 0	0	SISm0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)

<R>

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

(f) Serial data register mn (SDRmn)

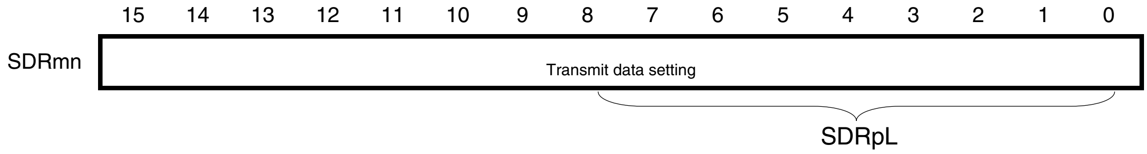
(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)
□ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-27. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (2/2)

(ii) During operation (SEm_n = 1) (lower 8 bits: SDRpL)



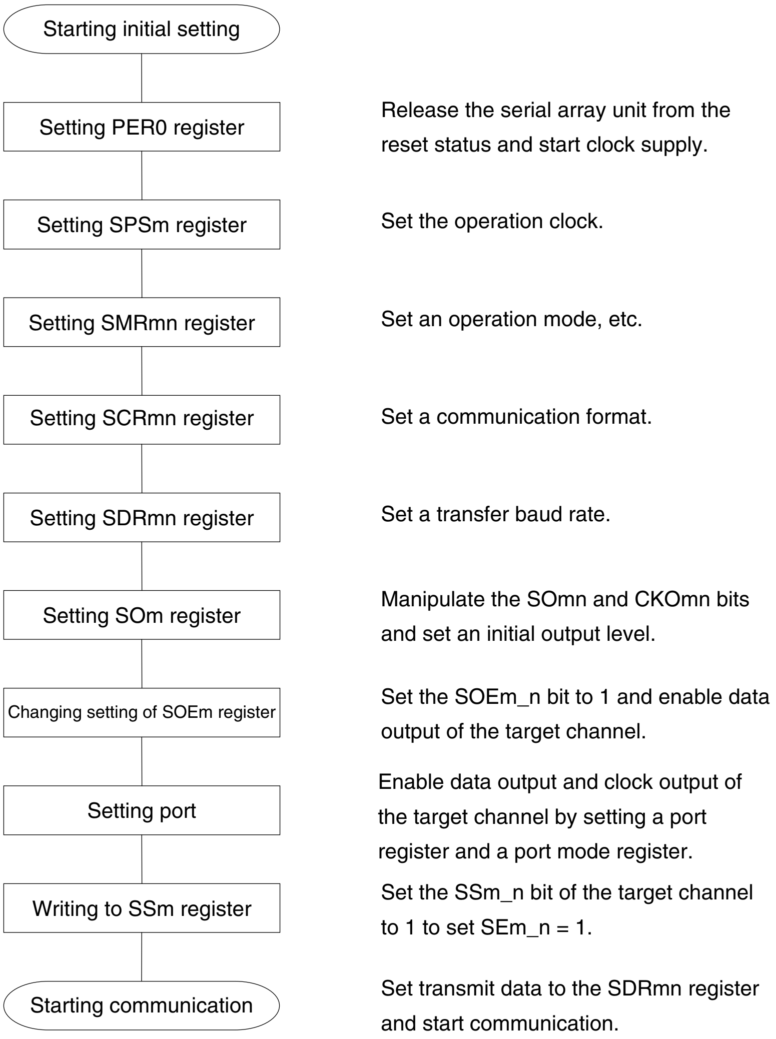
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

□ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

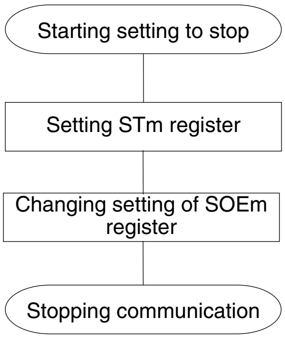
Figure 11-28. Initial Setting Procedure for Master Transmission


Starting initial setting
Setting PER0 register
Release the serial array unit from the reset status and start clock supply.
Setting SPSm register
Set the operation clock.
Setting SMRmn register
Set an operation mode, etc.
Setting SCRmn register
Set a communication format.
Setting SDRmn register
Set a transfer baud rate.
Setting SOm register
Manipulate the SOmn and CKOmn bits and set an initial output level.
Changing setting of SOEm register
Set the SOEm_n bit to 1 and enable data output of the target channel.
Setting port
Enable data output and clock output of the target channel by setting a port register and a port mode register.
Writing to SSm register
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.
Starting communication
Set transmit data to the SDRmn register and start communication.

Caution After setting the PER0 register 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, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 11-29. Procedure for Stopping Master Transmission



Write 1 to the STm_n bit of the target channel.

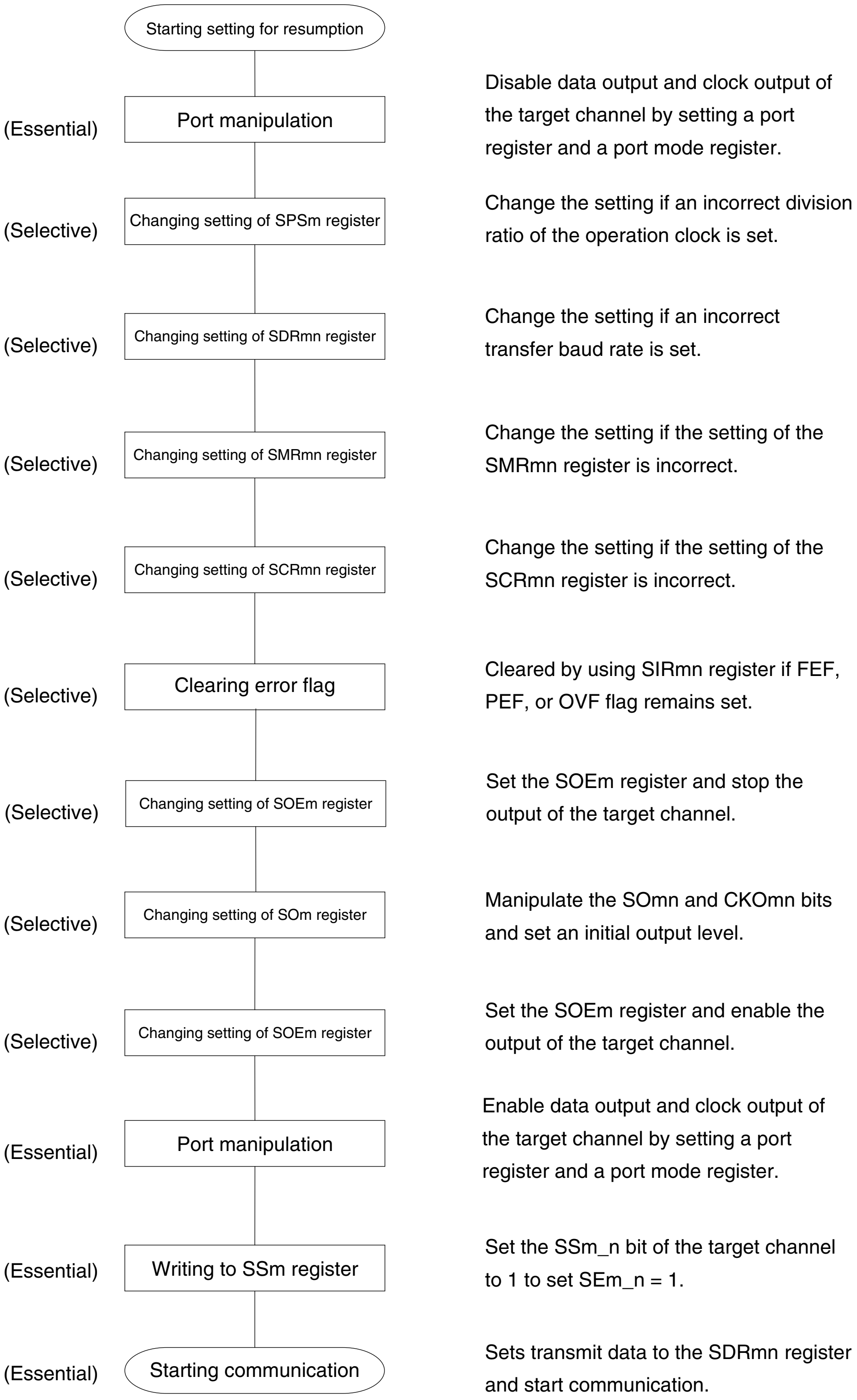
Clear the SOEm_n bit to 0 and stop the output of the target channel

Stop communication in midway.

Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-30 Procedure for Resuming Master Transmission**).

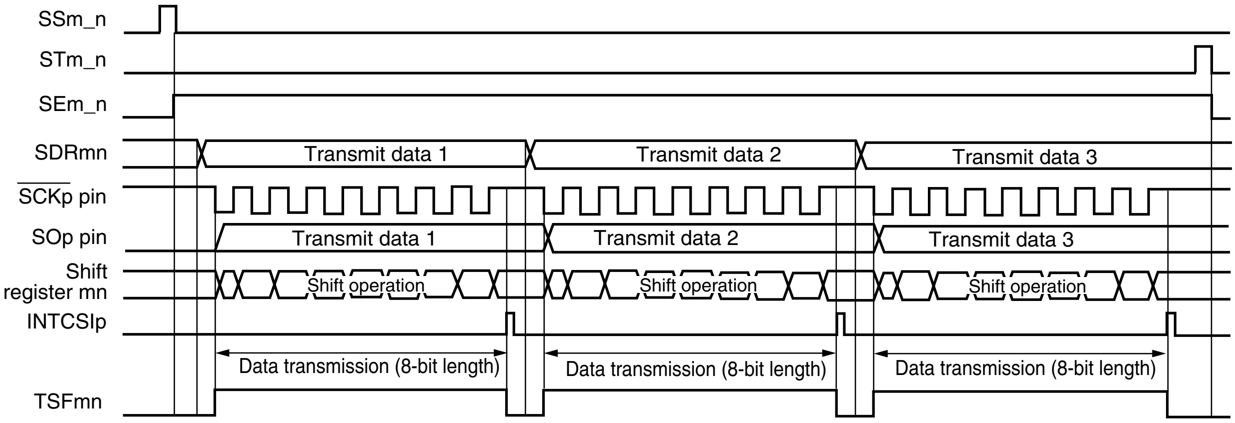
2. m: Unit number (m = 0, 1)

Figure 11-30. Procedure for Resuming Master Transmission


Starting setting for resumption
(Essential)
Port manipulation
Disable data output and clock output of the target channel by setting a port register and a port mode register.
(Selective)
Changing setting of SPSm register
Change the setting if an incorrect division ratio of the operation clock is set.
(Selective)
Changing setting of SDRmn register
Change the setting if an incorrect transfer baud rate is set.
(Selective)
Changing setting of SMRmn register
Change the setting if the setting of the SMRmn register is incorrect.
(Selective)
Changing setting of SCRmn register
Change the setting if the setting of the SCRmn register is incorrect.
(Selective)
Clearing error flag
Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.
(Selective)
Changing setting of SOEm register
Set the SOEm register and stop the output of the target channel.
(Selective)
Changing setting of SOm register
Manipulate the SOmn and CKOmn bits and set an initial output level.
(Selective)
Changing setting of SOEm register
Set the SOEm register and enable the output of the target channel.
(Essential)
Port manipulation
Enable data output and clock output of the target channel by setting a port register and a port mode register.
(Essential)
Writing to SSm register
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.
(Essential)
Starting communication
Sets transmit data to the SDRmn register and start communication.

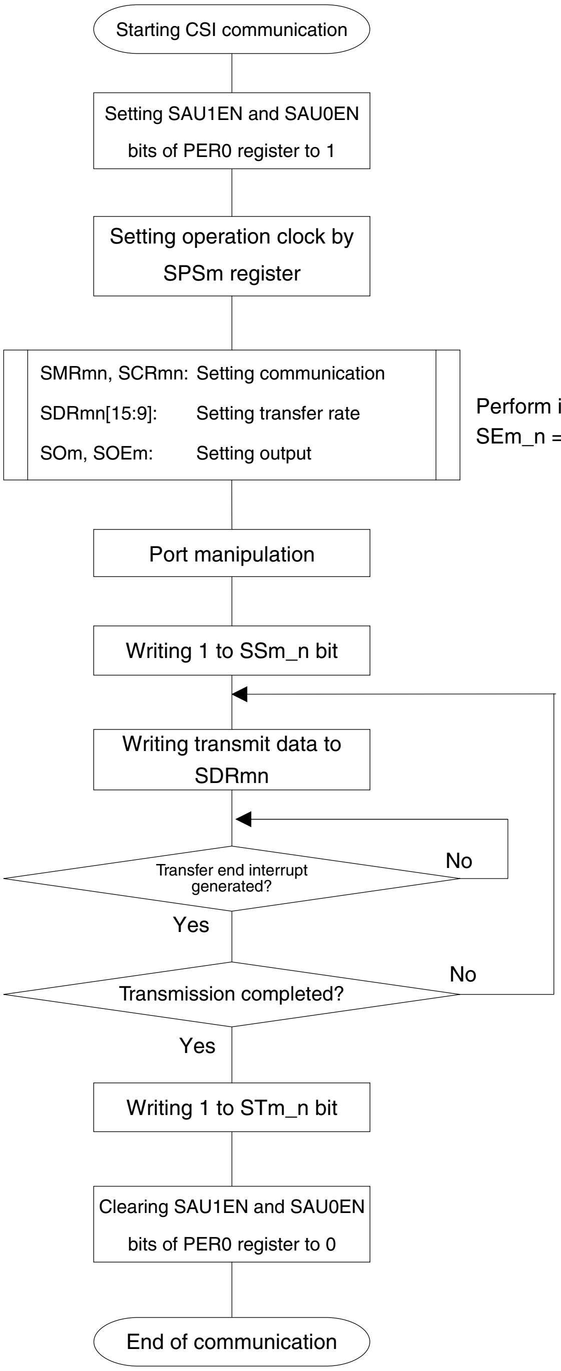
(3) Processing flow (in single-transmission mode)

Figure 11-31. Timing Chart of Master Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

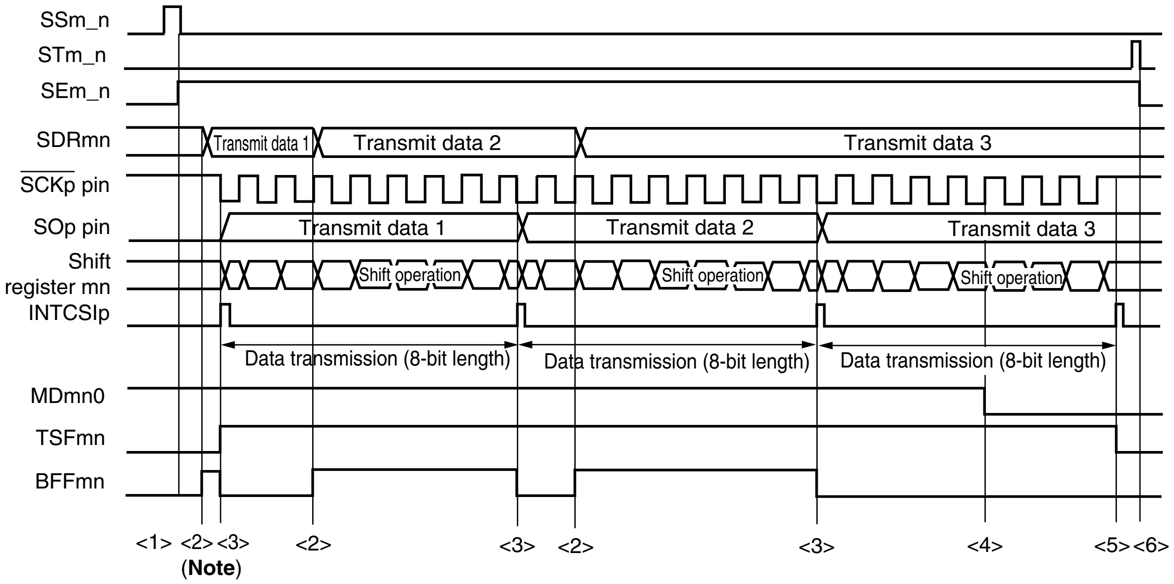
Figure 11-32. Flowchart of Master Transmission (in Single-Transmission Mode)


Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting transfer rate
SOm, SOEm: Setting output
Perform initial setting when SEm_n = 0.
Port manipulation
Writing 1 to SSm_n bit
Writing transmit data to SDRmn
Transfer end interrupt generated?
No
Yes
Transmission completed?
No
Yes
Writing 1 to STm_n bit
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 11-33. Timing Chart of Master 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.

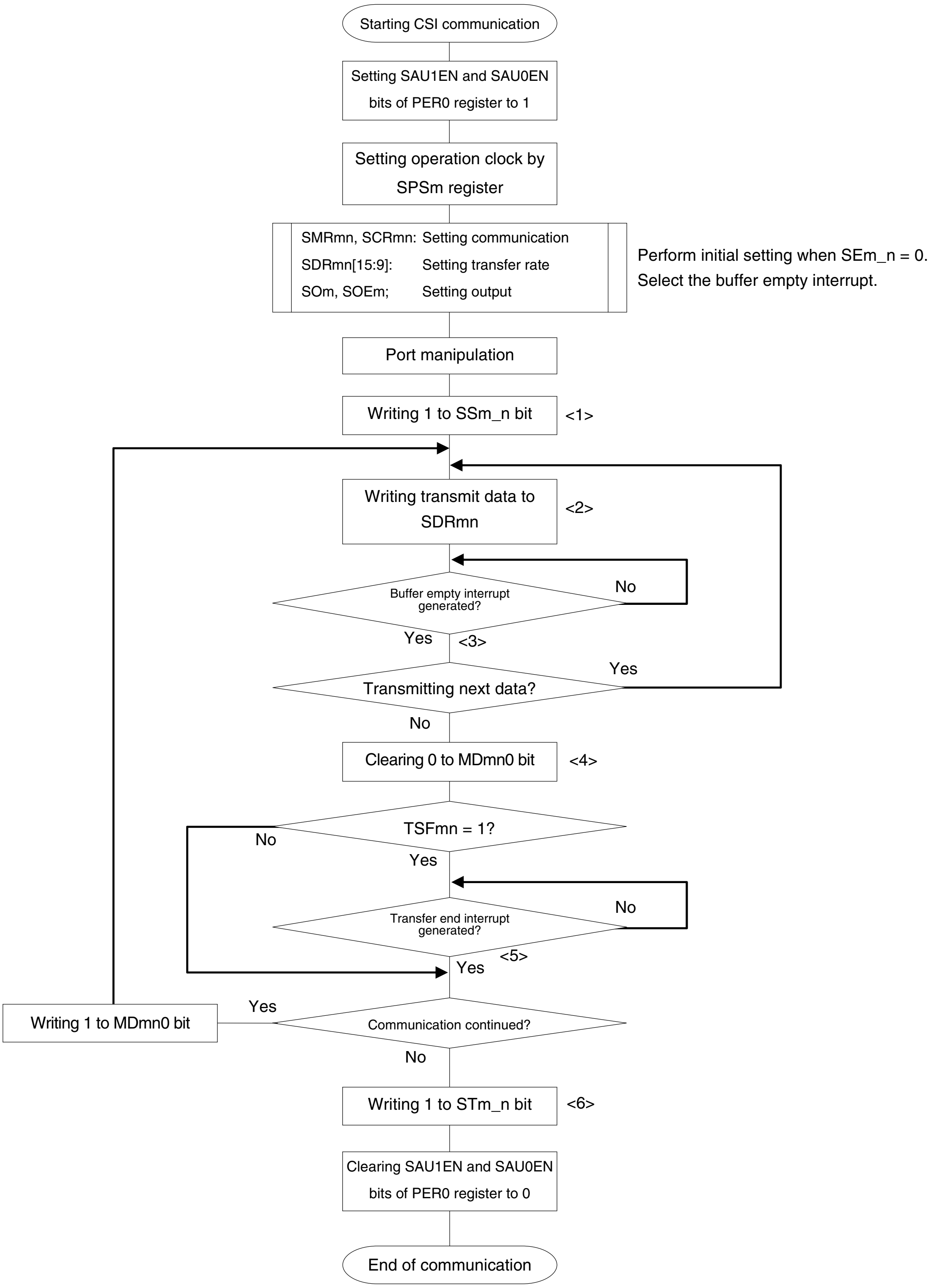
Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 11-34. Flowchart of Master Transmission (in Continuous Transmission Mode)

<R>

<R>


Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting transfer rate
SOm, SOEm; Setting output
Perform initial setting when SEm_n = 0.
Select the buffer empty interrupt.
Port manipulation
Writing 1 to SSm_n bit
<1>
Writing transmit data to SDRmn
<2>
Buffer empty interrupt generated?
No
Yes
<3>
Transmitting next data?
Yes
No
Clearing 0 to MDmn0 bit
<4>
TSFmn = 1?
No
Yes
Transfer end interrupt generated?
No
<5>
Yes
Writing 1 to MDmn0 bit
Yes
Communication continued?
No
Writing 1 to STm_n bit
<6>
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-33 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

11.5.2 Master reception

Master reception is that the 78K0R/Hx3 outputs a transfer clock and receives data from other device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	$\overline{\text{SCK00}}$, SI00	$\overline{\text{SCK01}}$, SI01	$\overline{\text{SCK10}}$, SI10	$\overline{\text{SCK11}}$, SI11
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11
	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 to 16 bits			
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz]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 29 ELECTRICAL SPECIFICATIONS**).

(1) Register setting

Figure 11-35. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	0/1	0/1	0	0	0	0	0	0	×	×

Communication starts when these bits are 1 if the data phase is forward (CKPmn = 0). If the phase is reversed (CKPmn = 1), communication starts when these bits are 0.

<R> **(b) Serial output enable register m (SOEm) …Set the receive target channel to 0.**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm															SOEm_1	SOEm_0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	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															SSm_1	SSm_0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	SCCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)

<R>

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn			PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0	DLSmn3	DLSmn2	DLSmn1	DLSmn0
	0	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0/1	0/1	0/1	0/1

(f) Serial data register mn (SDRmn)

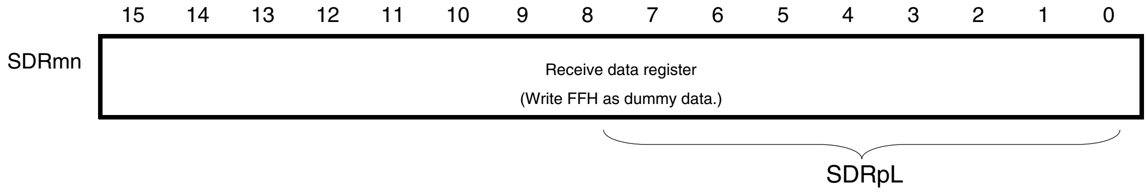
(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)
□: 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

Figure 11-35. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (2/2)

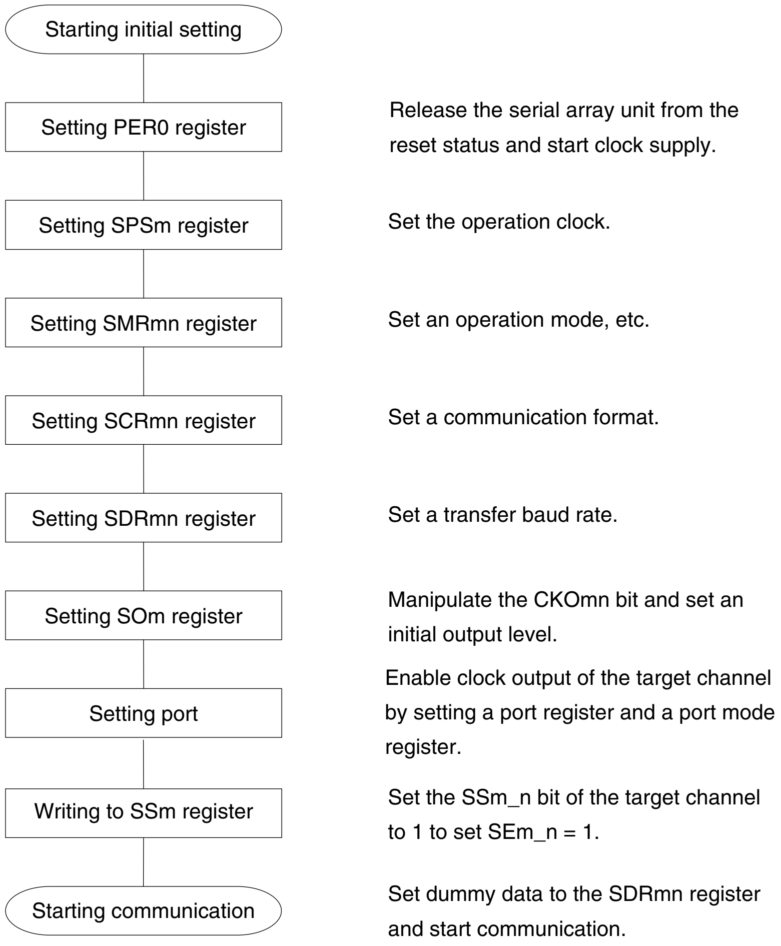
(ii) During operation (SEm_n = 1) (lower 8 bits: SDRpL)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)
□: Setting is fixed in the CSI master transmission mode, ▨: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

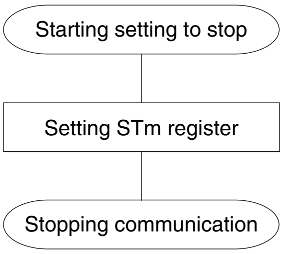
(2) Operation procedure

Figure 11-36. Initial Setting Procedure for Master Reception


Starting initial setting
Setting PER0 register
Release the serial array unit from the reset status and start clock supply.
Setting SPSm register
Set the operation clock.
Setting SMRmn register
Set an operation mode, etc.
Setting SCRmn register
Set a communication format.
Setting SDRmn register
Set a transfer baud rate.
Setting SOm register
Manipulate the CKOmn bit and set an initial output level.
Setting port
Enable clock output of the target channel by setting a port register and a port mode register.
Writing to SSm register
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.
Starting communication
Set dummy data to the SDRmn register and start communication.

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-37. Procedure for Stopping Master Reception

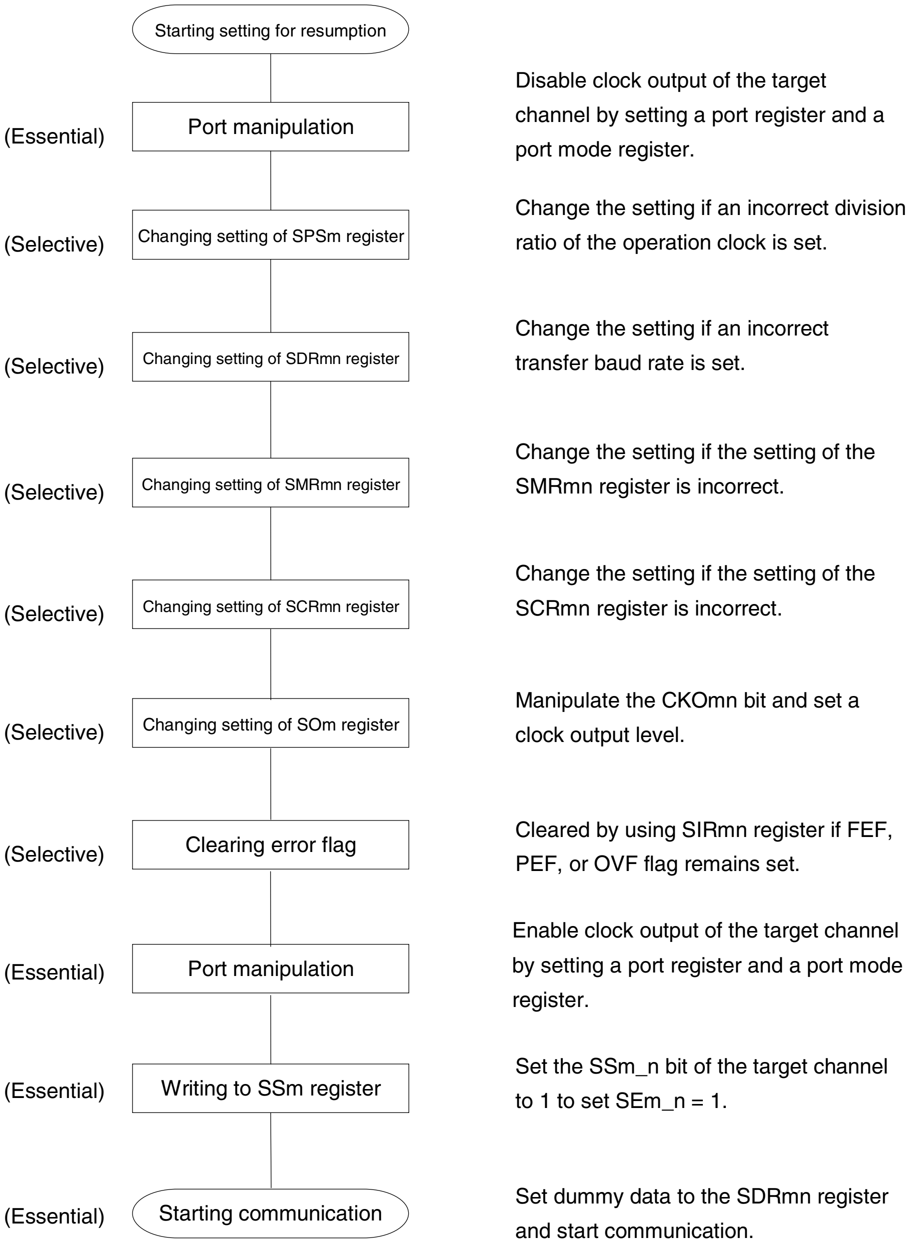


Write 1 to the STm_n bit of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-38 Procedure for Resuming Master Reception**).

Figure 11-38. Procedure for Resuming Master Reception



Disable clock output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if an incorrect transfer baud rate is set.

Change the setting if the setting of the SMRmn register is incorrect.

Change the setting if the setting of the SCRmn register is incorrect.

Manipulate the CKOmn bit and set a clock output level.

Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.

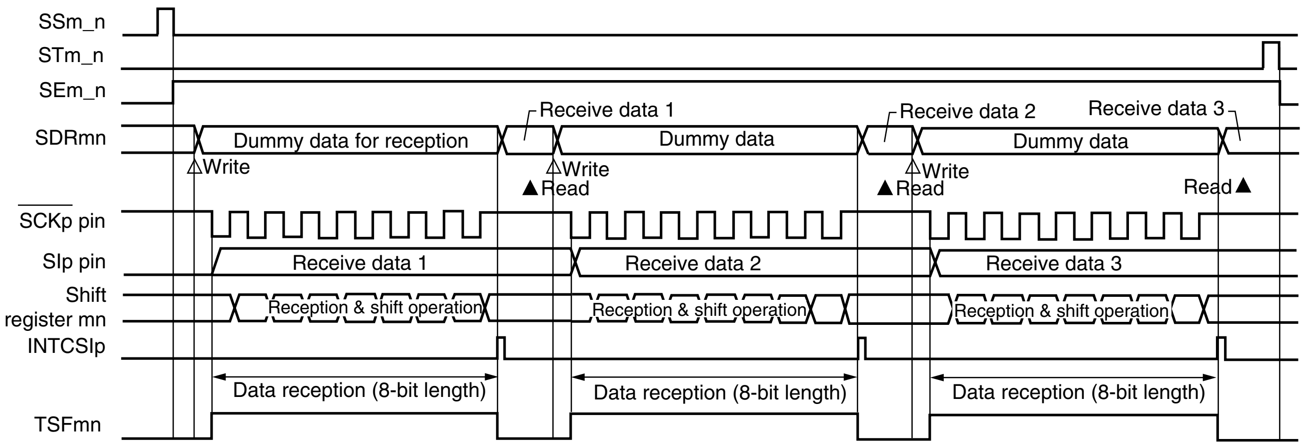
Enable clock output of the target channel by setting a port register and a port mode register.

Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Set dummy data to the SDRmn register and start communication.

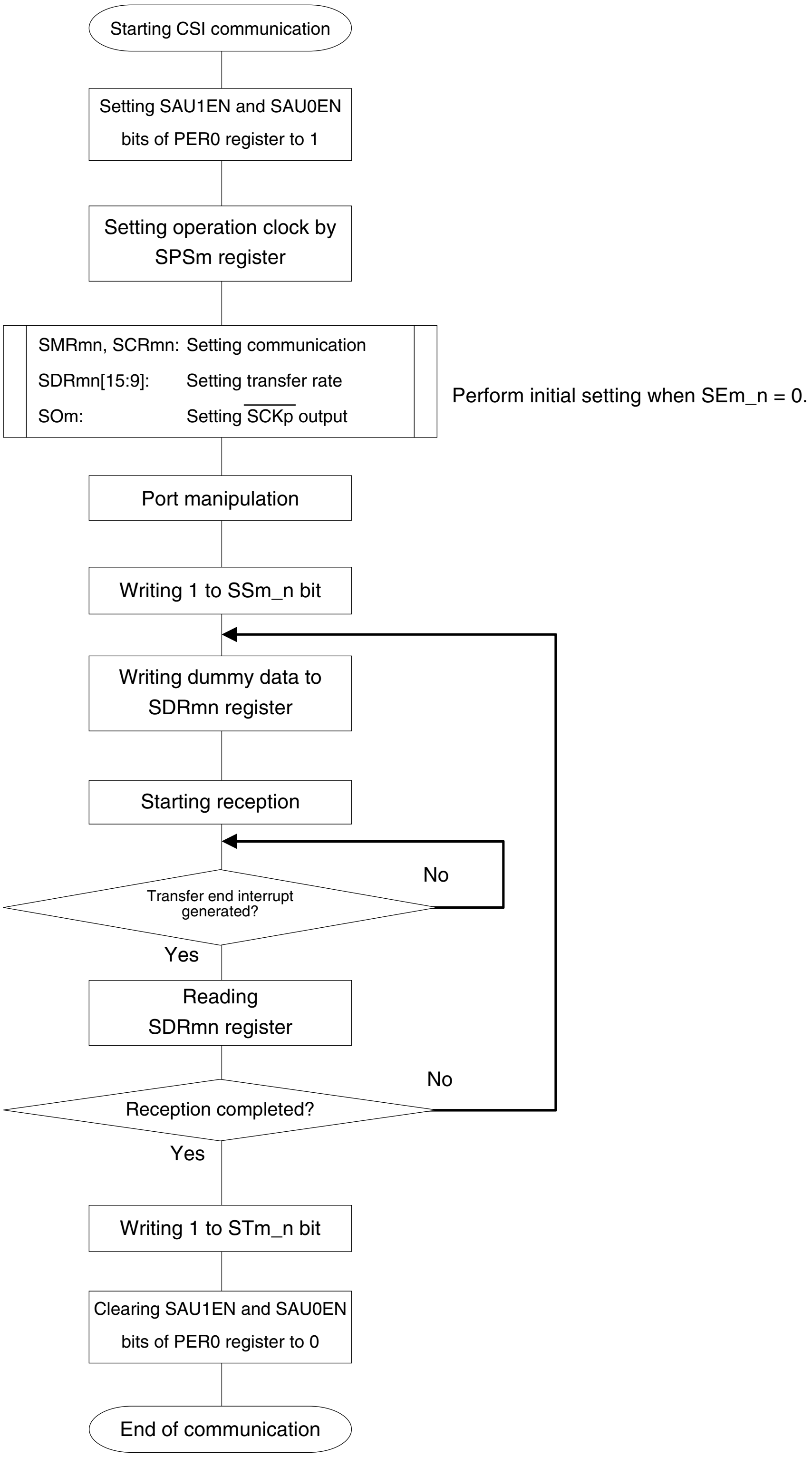
(3) Processing flow (in single-reception mode)

Figure 11-39. Timing Chart of Master Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

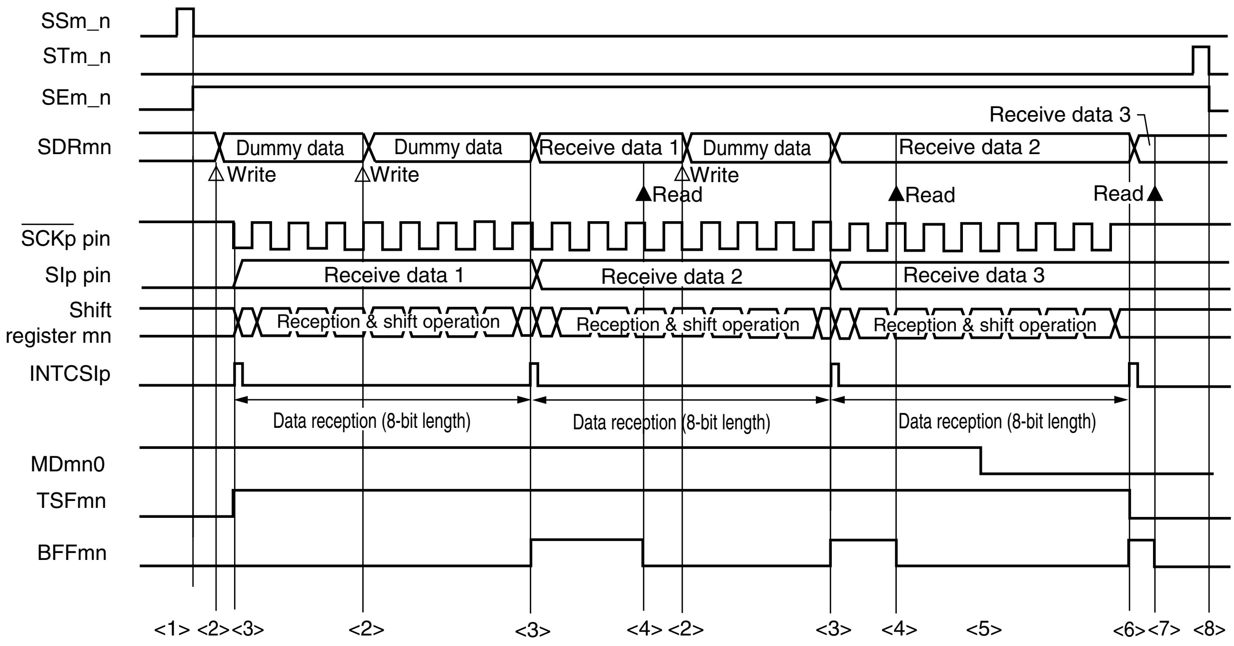
Figure 11-40. Flowchart of Master Reception (in Single-Reception Mode)


Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting transfer rate
SOm: Setting $\overline{\text{SCKp}}$ output
Perform initial setting when SEm_n = 0.
Port manipulation
Writing 1 to SSm_n bit
Writing dummy data to SDRmn register
Starting reception
Transfer end interrupt generated?
No
Yes
Reading SDRmn register
Reception completed?
No
Yes
Writing 1 to STm_n bit
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous reception mode)

Figure 11-41. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Caution **The MDmn0 bit can be rewritten even during operation.**
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-42 Flowchart of Master Reception (in Continuous Reception Mode)**.

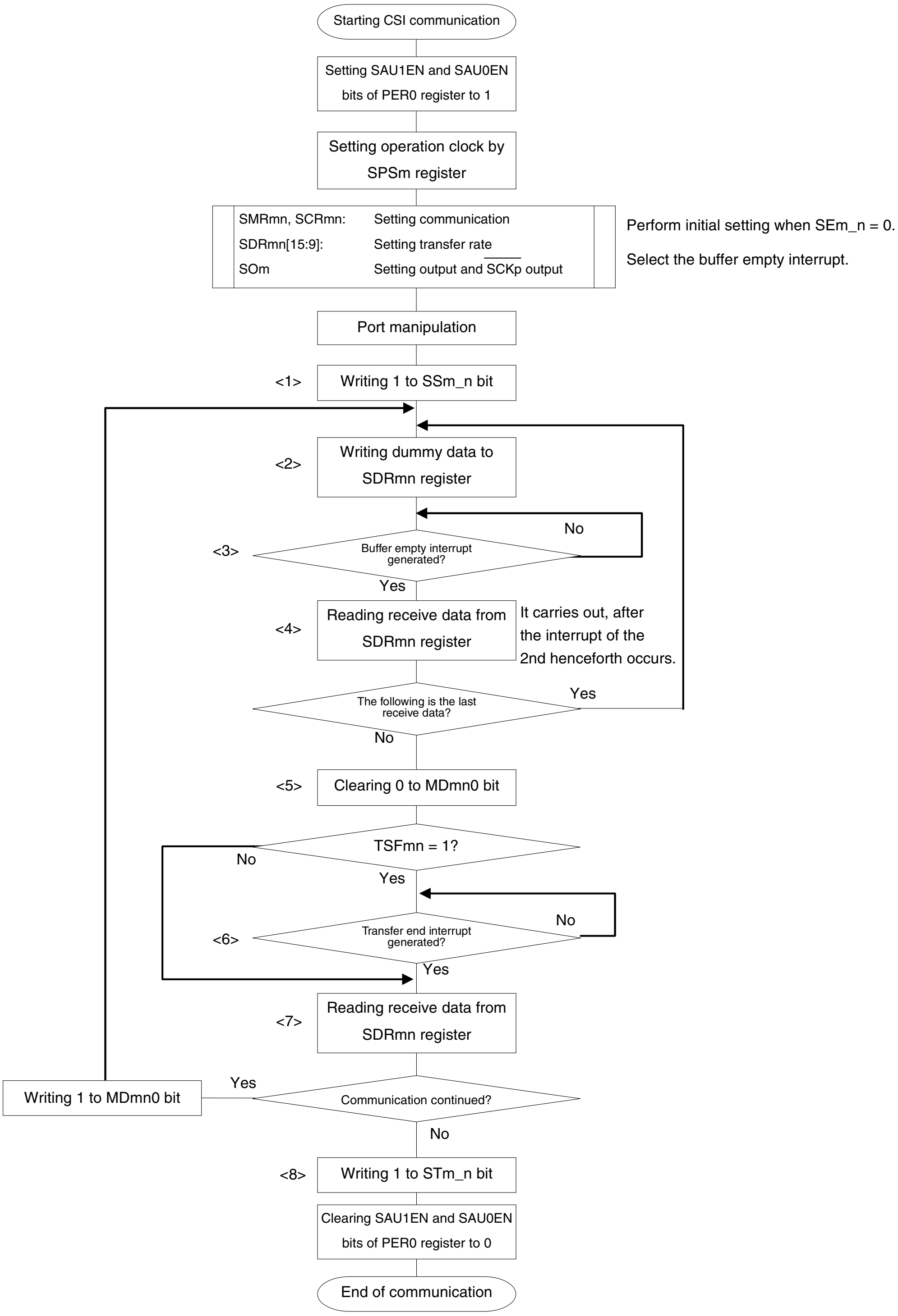
2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 11-42. Flowchart of Master Reception (in Continuous Reception Mode)

<R>

<R>

<R>


Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting transfer rate
SOm Setting output and SCKp output
Perform initial setting when SEm_n = 0.
Select the buffer empty interrupt.
Port manipulation
<1> Writing 1 to SSm_n bit
<2> Writing dummy data to SDRmn register
<3> Buffer empty interrupt generated?
No
Yes
<4> Reading receive data from SDRmn register
It carries out, after the interrupt of the 2nd henceforth occurs.
The following is the last receive data?
Yes
No
<5> Clearing 0 to MDmn0 bit
TSFmn = 1?
No
Yes
<6> Transfer end interrupt generated?
No
Yes
<7> Reading receive data from SDRmn register
Writing 1 to MDmn0 bit
Yes
Communication continued?
No
<8> Writing 1 to STm_n bit
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-41 Timing Chart of Master Reception (in Continuous Reception Mode)**.

11.5.3 Master transmission/reception

Master transmission/reception is that the 78K0R/Hx3 outputs a transfer clock and transmits/receives data to/from other device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	$\overline{\text{SCK00}}$, SI00, SO00	$\overline{\text{SCK01}}$, SI01, SO01	$\overline{\text{SCK10}}$, SI10, SO10	$\overline{\text{SCK11}}$, SI11, SO11
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11
	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 to 16 bits			
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz]Note f_{CLK}: System clock frequency			
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data input/output starts at the start of the operation of the serial clock. • DAPmn = 1: Data input/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 29 ELECTRICAL SPECIFICATIONS**).

(1) Register setting

Figure 11-43. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	0/1	0/1	0	0	0	0	0	0	0/1	0/1

Communication starts when these bits are 1 if the data phase is forward (CKPmn = 0). If the phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(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															SOEm_1	SOEm_0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	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															SSm_1	SSm_0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	SCCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)

<R>

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn			PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0	DLSmn3	DLSmn2	DLSmn1	DLSmn0
	1	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0/1	0/1	0/1	0/1

(f) Serial data register mn (SDRmn)

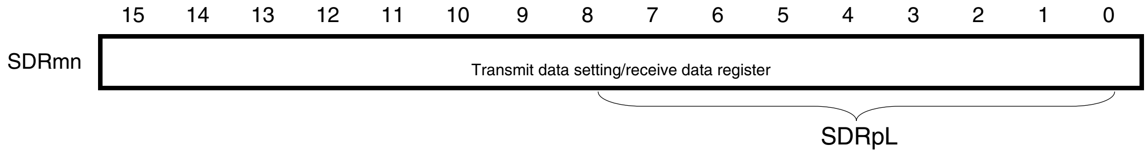
(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)
□ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-43. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (2/2)

(ii) During operation (SEm_n = 1) (lower 8 bits: SDRpL)



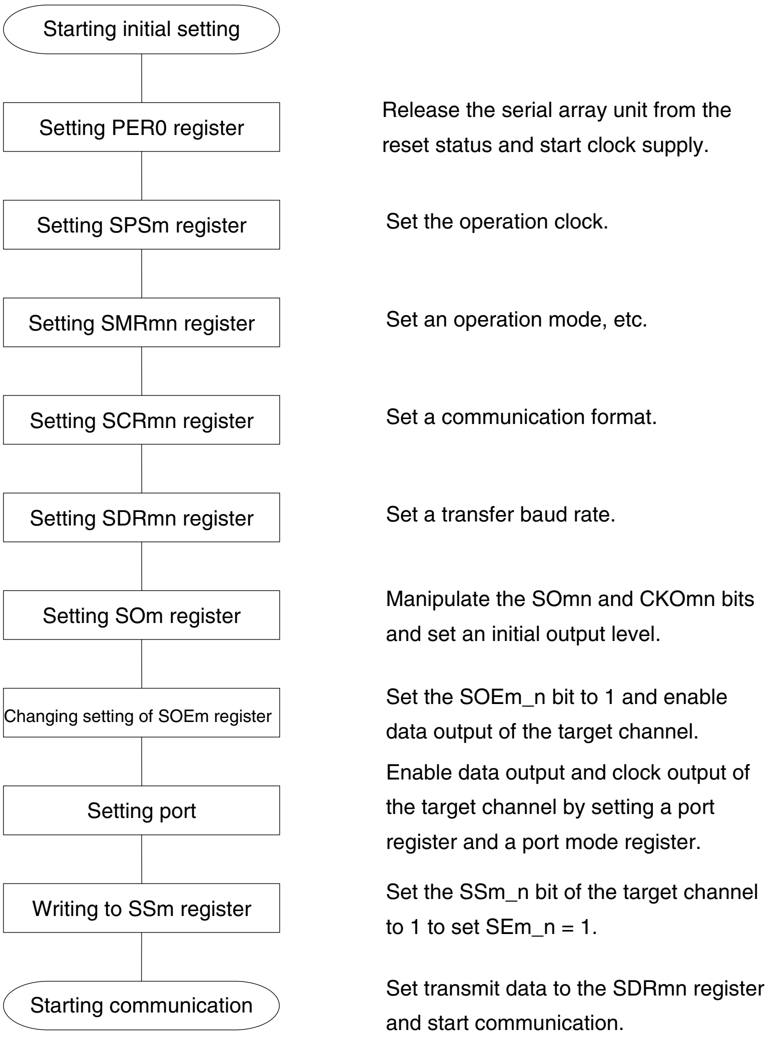
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

□ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-44. Initial Setting Procedure for Master Transmission/Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate.

Manipulate the SOmn and CKOmn bits and set an initial output level.

Set the SOEm_n bit to 1 and enable data output of the target channel.

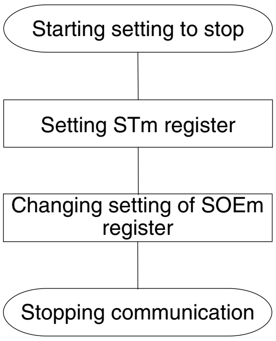
Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Set transmit data to the SDRmn register and start communication.

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-45. Procedure for Stopping Master Transmission/Reception



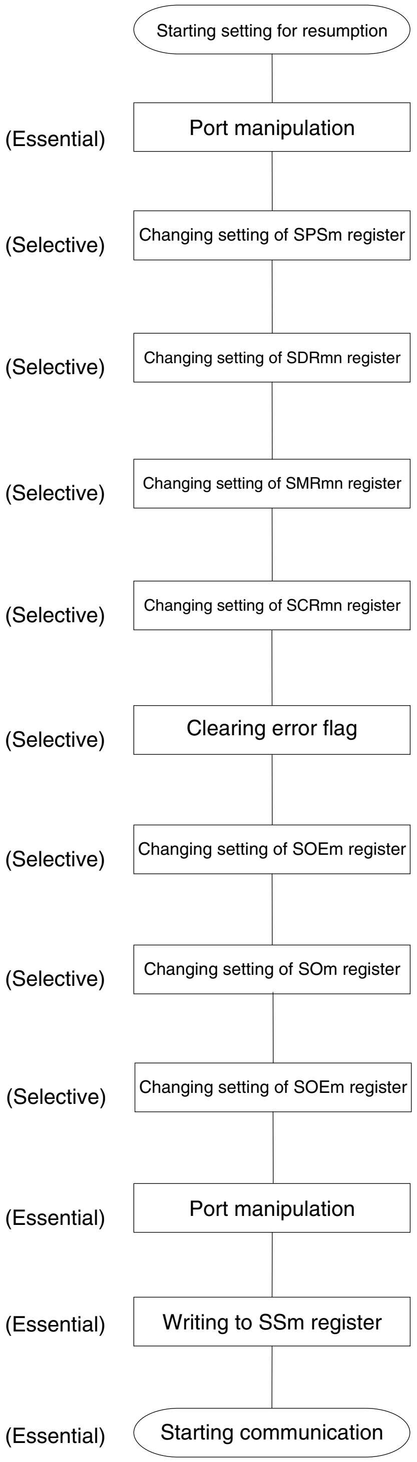
Write 1 to the STm_n bit of the target channel.

Clear the SOEm_n bit to 0 and stop the output of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-46 Procedure for Resuming Master Transmission/Reception**).

Figure 11-46. Procedure for Resuming Master Transmission/Reception


Starting setting for resumption
(Essential)
Port manipulation
(Selective)
Changing setting of SPSm register
(Selective)
Changing setting of SDRmn register
(Selective)
Changing setting of SMRmn register
(Selective)
Changing setting of SCRmn register
(Selective)
Clearing error flag
(Selective)
Changing setting of SOEm register
(Selective)
Changing setting of SOm register
(Selective)
Changing setting of SOEm register
(Essential)
Port manipulation
(Essential)
Writing to SSm register
(Essential)
Starting communication

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if an incorrect transfer baud rate is set.

Change the setting if the setting of the SMRmn register is incorrect.

Change the setting if the setting of the SCRmn register is incorrect.

Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.

Set the SOEm register and stop the output of the target channel.

Manipulate the SOmn and CKOmn bits and set an initial output level.

Set the SOEm register and enable data output of the target channel.

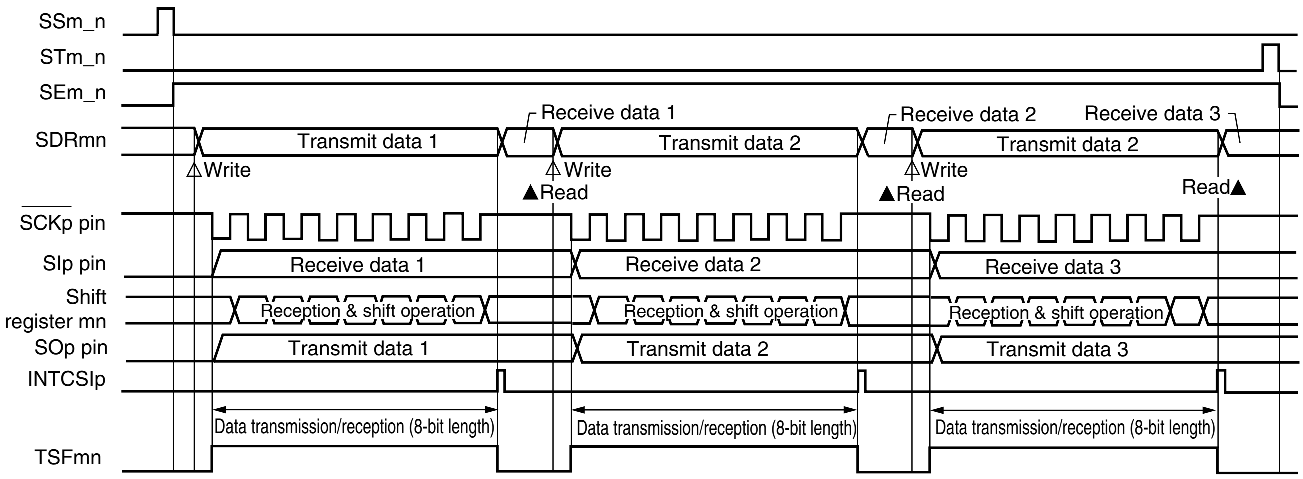
Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Sets transmit data to the SDRmn register and start communication.

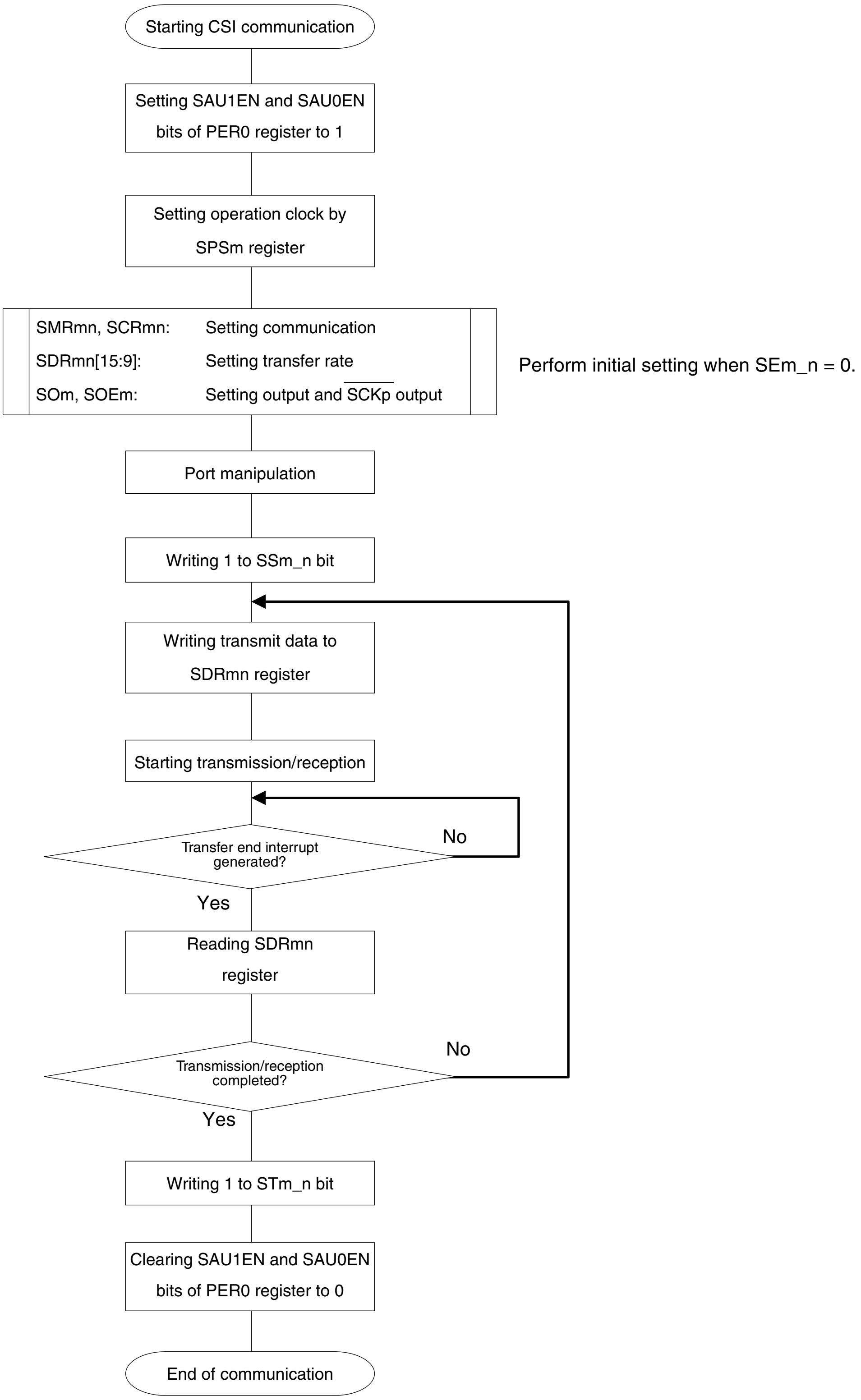
(3) Processing flow (in single-transmission/reception mode)

Figure 11-47. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

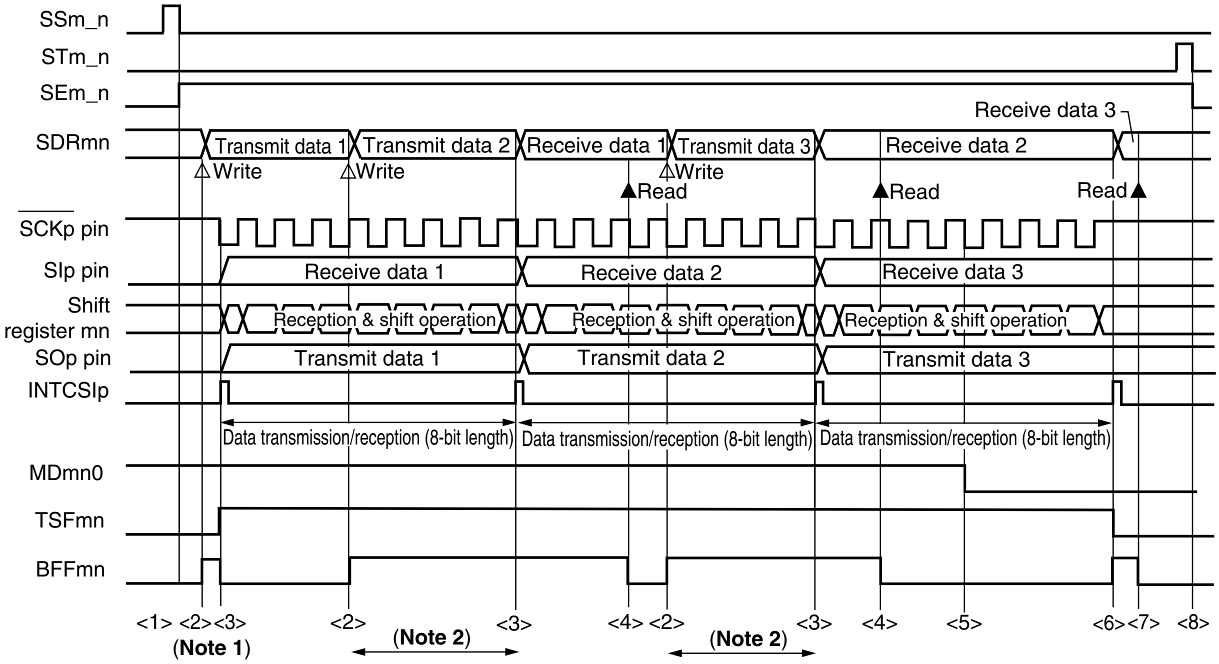
Figure 11-48. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)


Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting transfer rate
SOm, SOEm: Setting output and $\overline{\text{SCKp}}$ output
Perform initial setting when SEm_n = 0.
Port manipulation
Writing 1 to SSm_n bit
Writing transmit data to SDRmn register
Starting transmission/reception
Transfer end interrupt generated?
No
Yes
Reading SDRmn register
Transmission/reception completed?
No
Yes
Writing 1 to STm_n bit
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-49. 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.

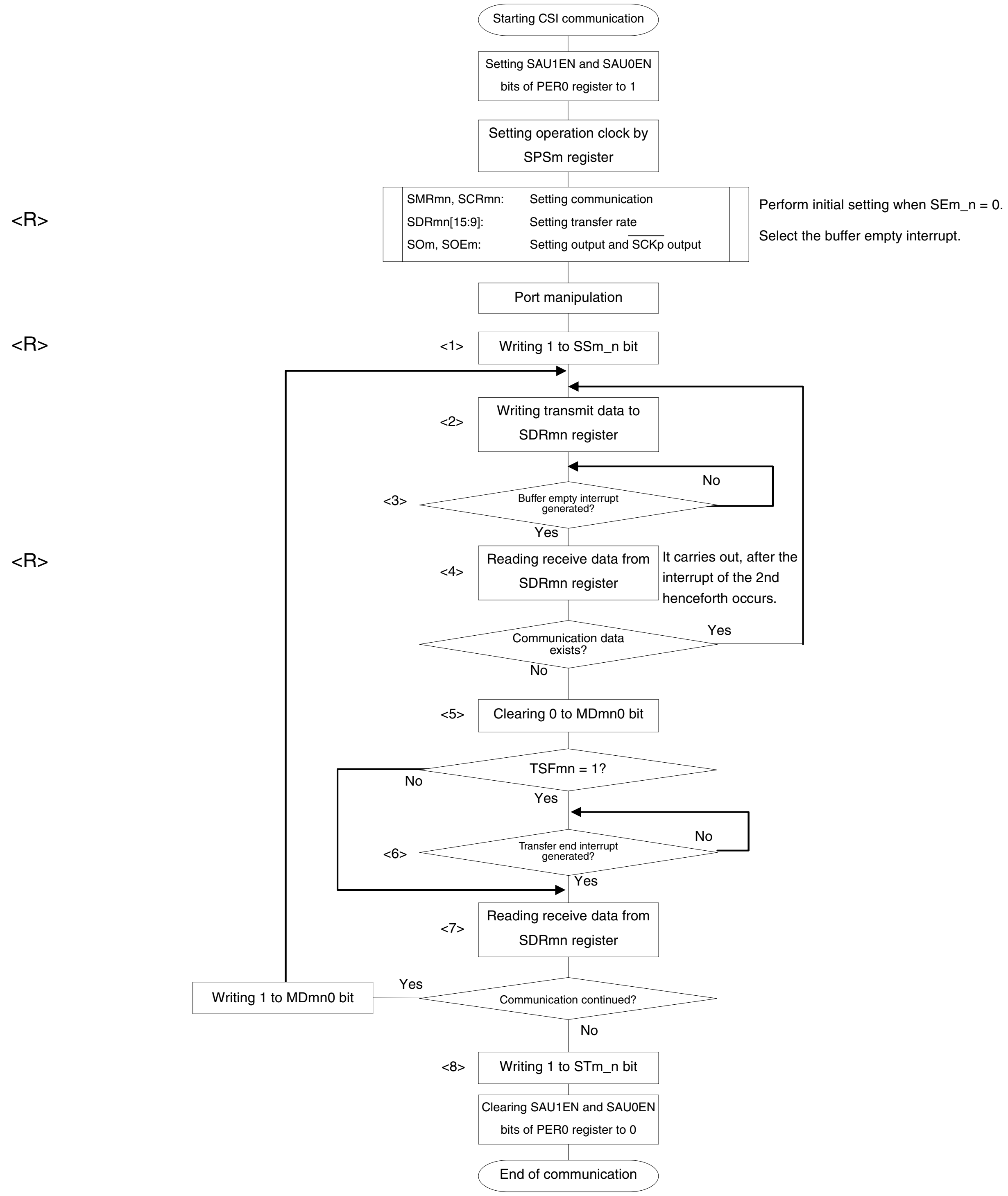
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 11-50 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 11-50. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)


Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
<R>
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting transfer rate
SOm, SOEm: Setting output and SCKp output
Perform initial setting when SEm_n = 0.
Select the buffer empty interrupt.
Port manipulation
<R>
<1>
Writing 1 to SSm_n bit
<2>
Writing transmit data to SDRmn register
<3>
Buffer empty interrupt generated?
No
Yes
<R>
<4>
Reading receive data from SDRmn register
It carries out, after the interrupt of the 2nd henceforth occurs.
Communication data exists?
Yes
No
<5>
Clearing 0 to MDmn0 bit
TSFmn = 1?
No
Yes
<6>
Transfer end interrupt generated?
No
Yes
<7>
Reading receive data from SDRmn register
Writing 1 to MDmn0 bit
Yes
Communication continued?
No
<8>
Writing 1 to STm_n bit
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

Caution **After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.**

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-49 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

11.5.4 Slave transmission

Slave transmission is that the 78K0R/Hx3 transmits data to another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	$\overline{\text{SCK00}}$, SO00	$\overline{\text{SCK01}}$, SO01	$\overline{\text{SCK10}}$, SO10	$\overline{\text{SCK11}}$, SO11
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11
	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 to 16 bits			
Transfer rate	Max. f_{MCK}/6 [Hz]$^{\text{Notes 1, 2}}$			
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 pins $\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$, and $\overline{\text{SCK11}}$ is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-51. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	×	×	0	0	0	0	0	0	0/1	0/1

(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															SOEm_1	SOEm_0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	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															SSm_1	SSm_0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	SCCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)

<R>

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn			PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0	DLSmn3	DLSmn2	DLSmn1	DLSmn0
	1	0	0/1	0/1	0	0	0	0	0/1	0	0	0	0/1	0/1	0/1	0/1

(f) Serial data register mn (SDRmn)

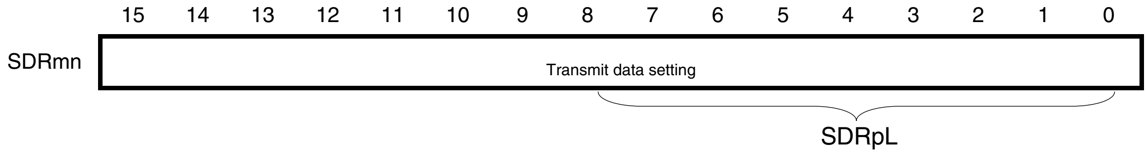
(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)
□ : 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

Figure 11-51. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (2/2)

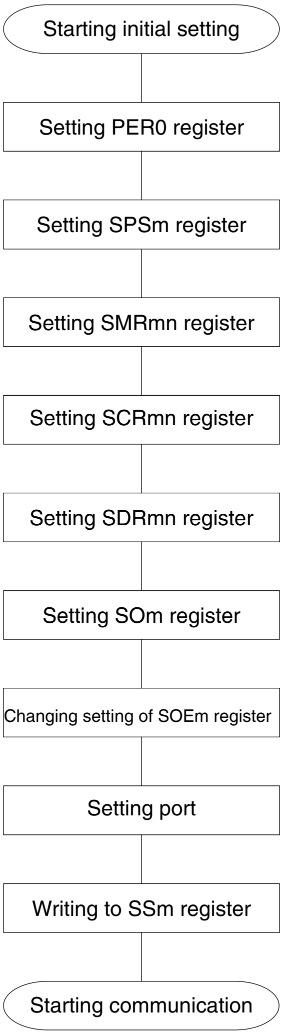
(ii) During operation (SEm_n = 1) (lower 8 bits: SDRpL)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)
□ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-52. Initial Setting Procedure for Slave Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to any value for setting the baud rate.

Manipulate the SOmn bit and set an initial output level.

Set the SOEm_n bit to 1 and enable data output of the target channel.

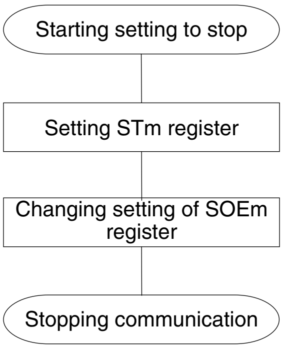
Enable data output of the target channel by setting a port register and a port mode register.

Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Set transmit data to the SDRmn register and wait for a clock from the master.

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-53. Procedure for Stopping Slave Transmission



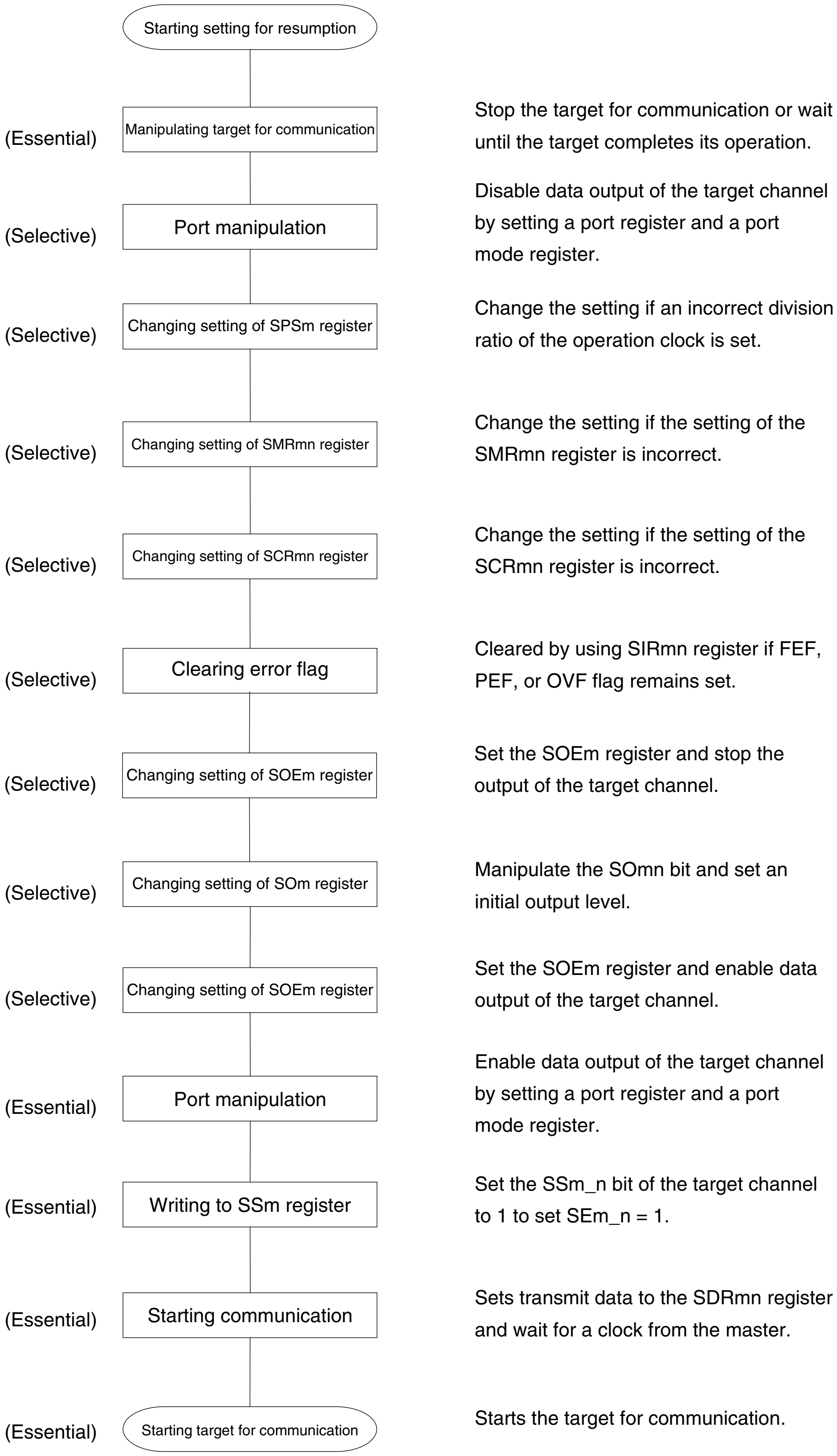
Write 1 to the STm_n bit of the target channel.

Clear the SOEm_n bit to 0 and stop the output of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-54 Procedure for Resuming Slave Transmission**).

Figure 11-54. Procedure for Resuming Slave Transmission


Starting setting for resumption
(Essential)
Manipulating target for communication
(Selective)
Port manipulation
(Selective)
Changing setting of SPSm register
(Selective)
Changing setting of SMRmn register
(Selective)
Changing setting of SCRmn register
(Selective)
Clearing error flag
(Selective)
Changing setting of SOEm register
(Selective)
Changing setting of SOm register
(Selective)
Changing setting of SOEm register
(Essential)
Port manipulation
(Essential)
Writing to SSm register
(Essential)
Starting communication
(Essential)
Starting target for communication

Stop the target for communication or wait until the target completes its operation.

Disable data output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if the setting of the SMRmn register is incorrect.

Change the setting if the setting of the SCRmn register is incorrect.

Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.

Set the SOEm register and stop the output of the target channel.

Manipulate the SOmn bit and set an initial output level.

Set the SOEm register and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

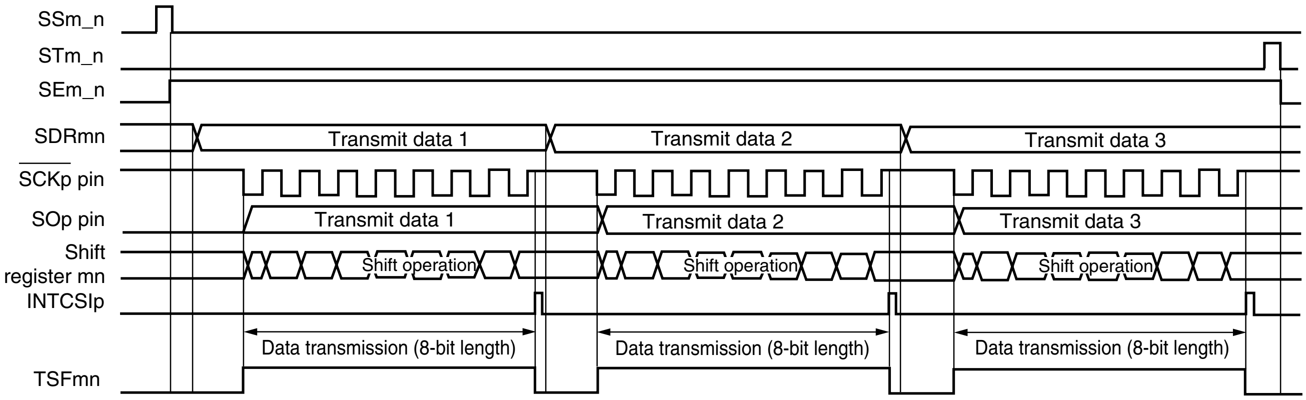
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Sets transmit data to the SDRmn register and wait for a clock from the master.

Starts the target for communication.

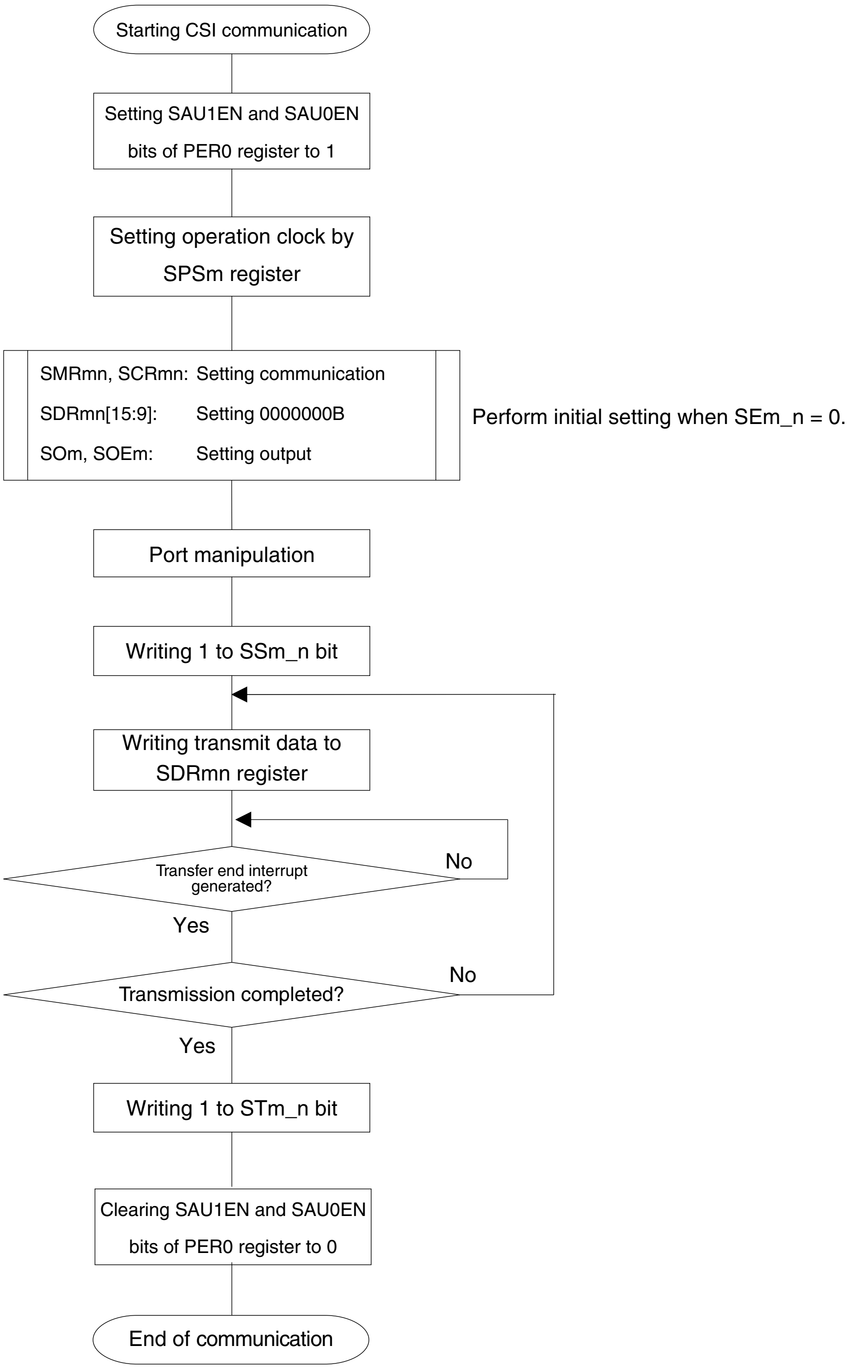
(3) Processing flow (in single-transmission mode)

Figure 11-55. Timing Chart of Slave Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

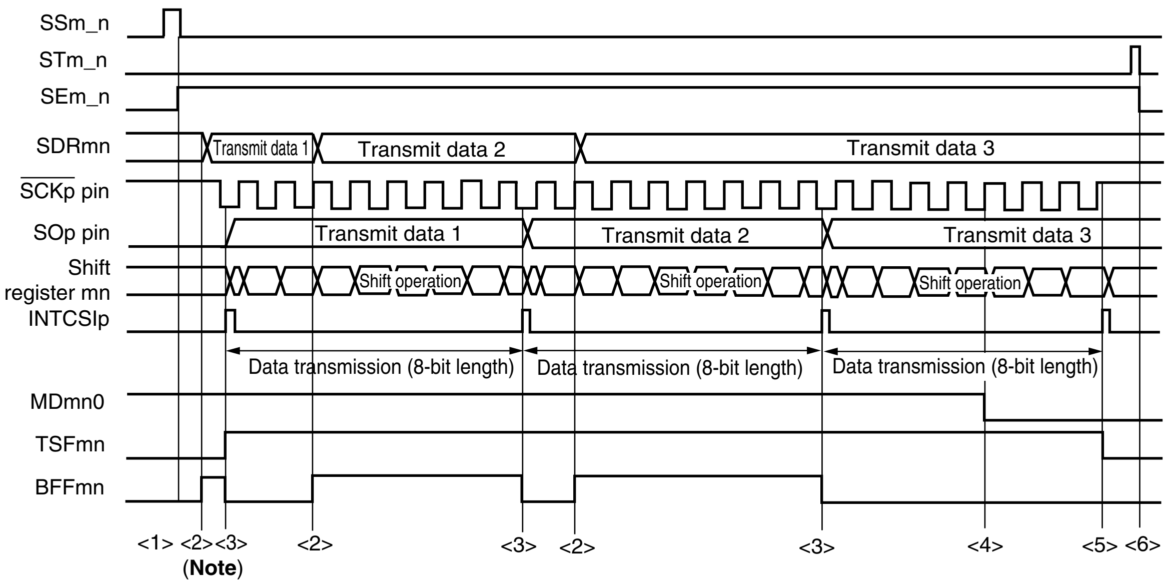
Figure 11-56. Flowchart of Slave Transmission (in Single-Transmission Mode)


Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting 0000000B
SOm, SOEm: Setting output
Perform initial setting when SEm_n = 0.
Port manipulation
Writing 1 to SSm_n bit
Writing transmit data to SDRmn register
Transfer end interrupt generated?
No
Yes
Transmission completed?
No
Yes
Writing 1 to STm_n bit
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 11-57. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



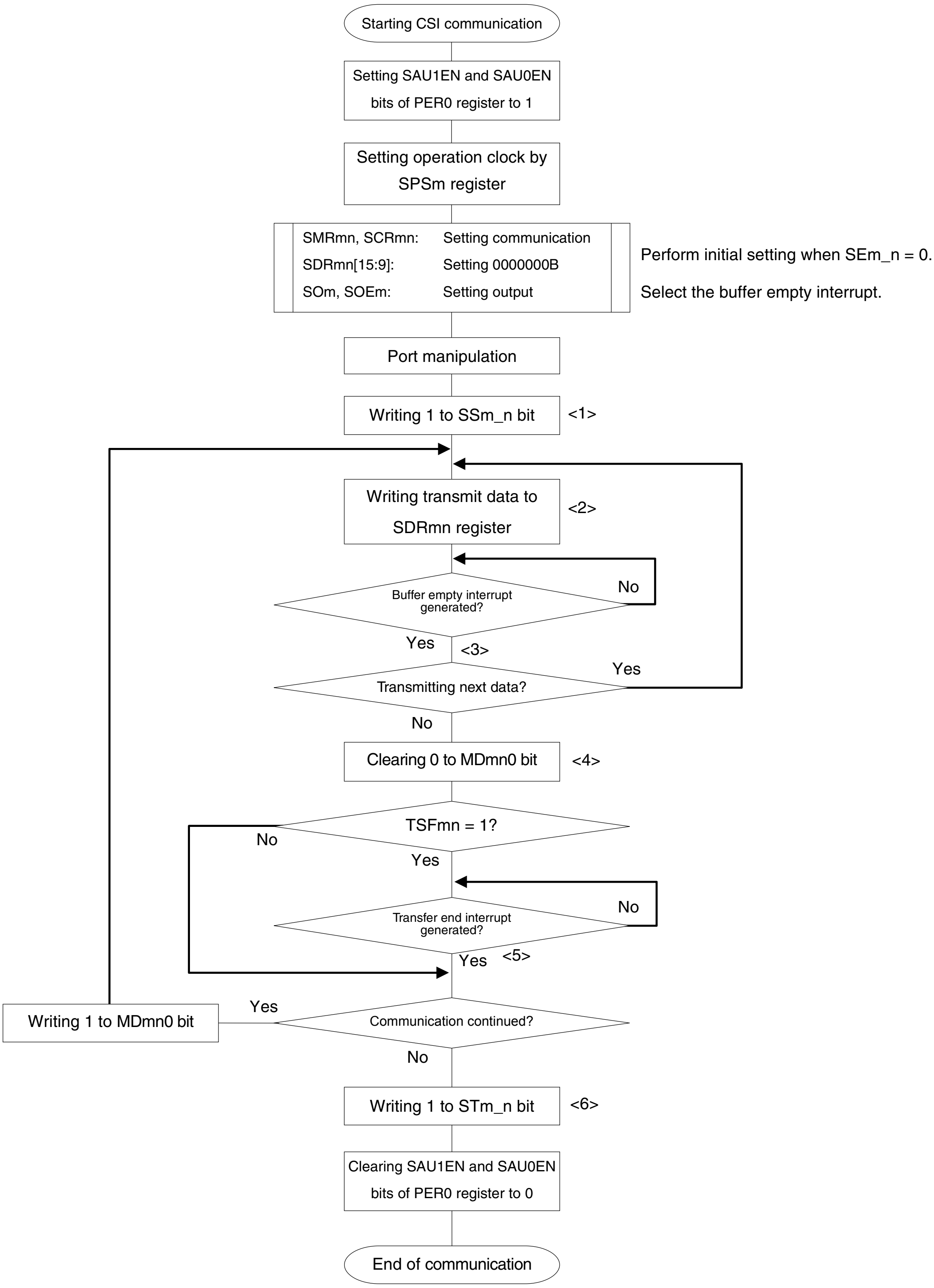
Note When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

Caution The MDmn0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Figure 11-58. Flowchart of Slave Transmission (in Continuous Transmission Mode)

<R>

<R>


Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting 0000000B
SOm, SOEm: Setting output
Perform initial setting when SEm_n = 0.
Select the buffer empty interrupt.
Port manipulation
Writing 1 to SSm_n bit
<1>
Writing transmit data to SDRmn register
<2>
Buffer empty interrupt generated?
No
Yes
<3>
Transmitting next data?
Yes
No
Clearing 0 to MDmn0 bit
<4>
TSFmn = 1?
No
Yes
Transfer end interrupt generated?
No
Yes
<5>
Writing 1 to MDmn0 bit
Yes
Communication continued?
No
Writing 1 to STm_n bit
<6>
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-57 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

11.5.5 Slave reception

Slave reception is that the 78K0R/Hx3 receives data from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	$\overline{\text{SCK00}}$, SI00	$\overline{\text{SCK01}}$, SI01	$\overline{\text{SCK10}}$, SI10	$\overline{\text{SCK11}}$, SI11
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 to 16 bits			
Transfer rate	Max. $f_{MCK}/6$ [Hz][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 pins $\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$, and $\overline{\text{SCK11}}$ is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-59. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)

(a) Serial output register m (SOm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 ×

<R> **(b) Serial output enable register m (SOEm) …Set the receive target channel to 0.**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm_1 0/1	SOEm_0 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	0	0	SSm_1 0/1	SSm_0 0/1

(d) 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	SCCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

Interrupt sources of channel n
0: Transfer end interrupt

(e) Serial communication operation setting register mn (SCRmn)

<R>

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

(f) Serial data register mn (SDRmn)

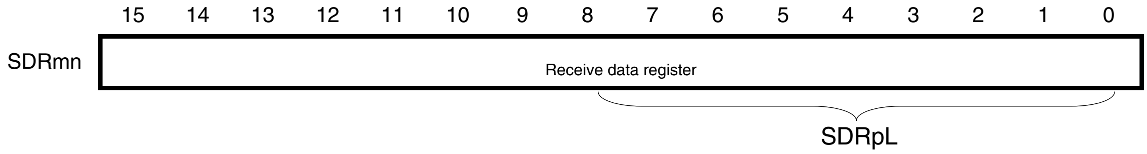
(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)
□ : 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

Figure 11-59. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (2/2)

(ii) During operation (SEm_n = 1) (lower 8 bits: SDRpL)



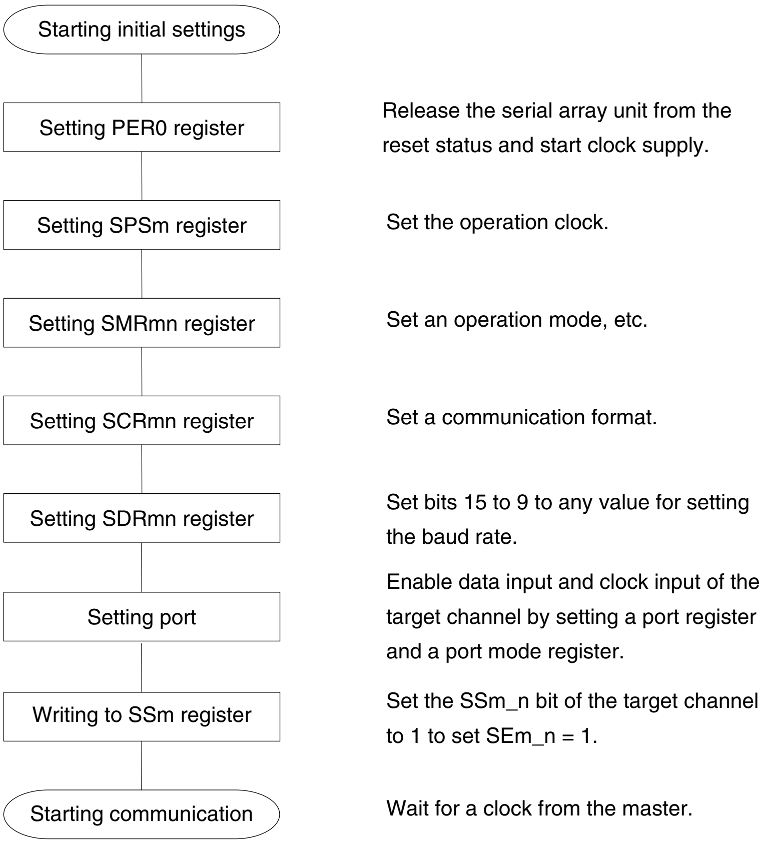
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

□ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-60. Initial Setting Procedure for Slave Reception

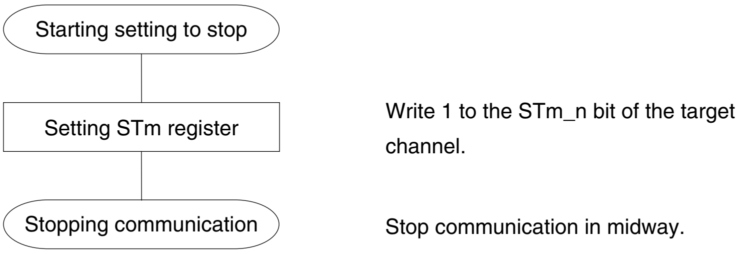


Starting initial settings
Setting PER0 register
Release the serial array unit from the reset status and start clock supply.
Setting SPSm register
Set the operation clock.
Setting SMRmn register
Set an operation mode, etc.
Setting SCRmn register
Set a communication format.
Setting SDRmn register
Set bits 15 to 9 to any value for setting the baud rate.
Setting port
Enable data input and clock input of the target channel by setting a port register and a port mode register.
Writing to SSm register
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.
Starting communication
Wait for a clock from the master.

<R>

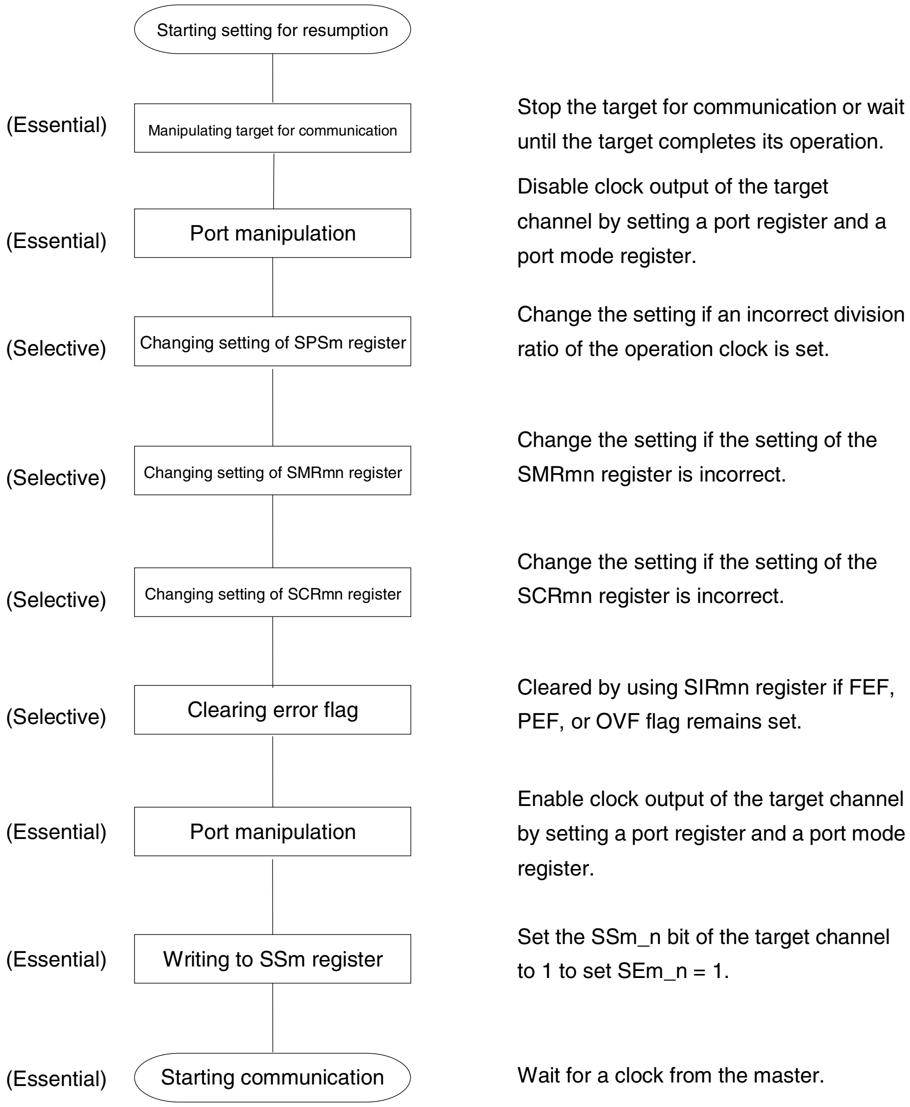
Cautions After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-61. Procedure for Stopping Slave Reception



Starting setting to stop
Setting STm register
Write 1 to the STm_n bit of the target channel.
Stopping communication
Stop communication in midway.

Figure 11-62. Procedure for Resuming Slave Reception



Stop the target for communication or wait until the target completes its operation.

Disable clock output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if the setting of the SMRmn register is incorrect.

Change the setting if the setting of the SCRmn register is incorrect.

Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.

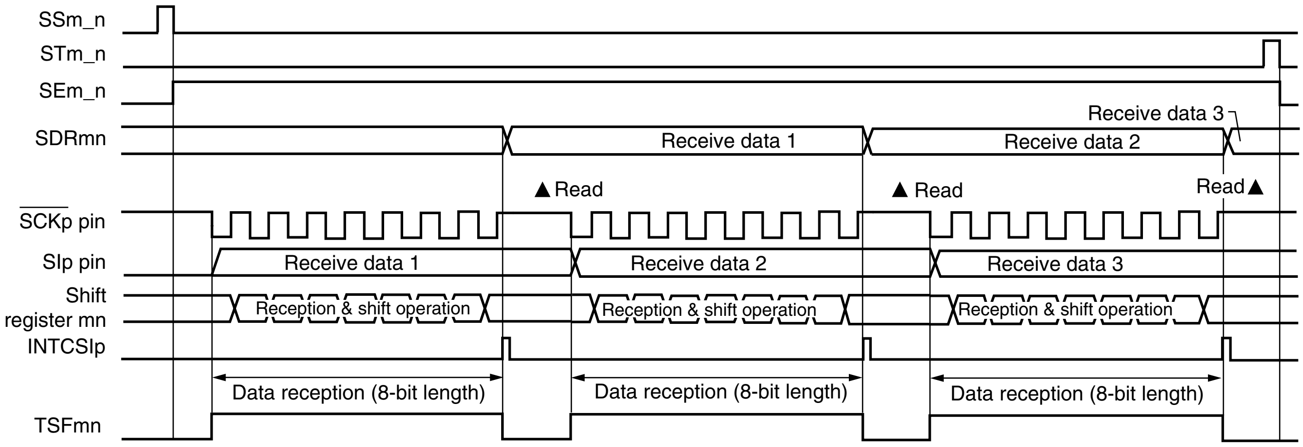
Enable clock output of the target channel by setting a port register and a port mode register.

Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Wait for a clock from the master.

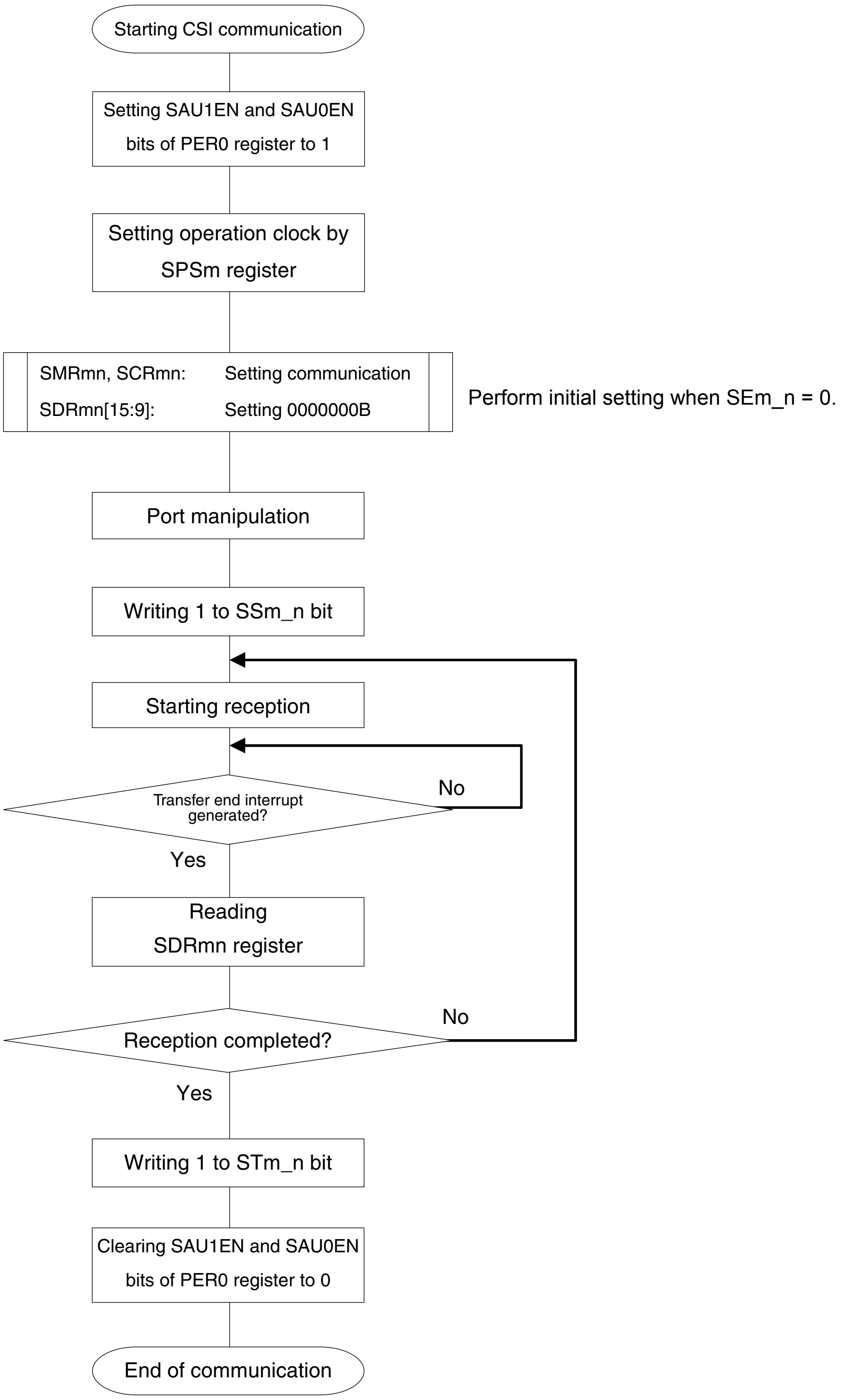
(3) Processing flow (in single-reception mode)

Figure 11-63. Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)


SSm_n
STm_n
SEm_n
SDRmn
Receive data 1
Receive data 2
Receive data 3
Read
Read
Read
SCKp pin
SIp pin
Receive data 1
Receive data 2
Receive data 3
Shift register mn
Reception & shift operation
Reception & shift operation
Reception & shift operation
INTCSIp
Data reception (8-bit length)
Data reception (8-bit length)
Data reception (8-bit length)
TSFmn

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 11-64. Flowchart of Slave Reception (in Single-Reception Mode)


Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting 0000000B
Perform initial setting when SEm_n = 0.
Port manipulation
Writing 1 to SSm_n bit
Starting reception
Transfer end interrupt generated?
No
Yes
Reading SDRmn register
Reception completed?
No
Yes
Writing 1 to STm_n bit
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

Caution **After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.**

11.5.6 Slave transmission/reception

Slave transmission/reception is that the 78K0R/Hx3 transmits/receives data to/from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	$\overline{\text{SCK00}}$, SI00, SO00	$\overline{\text{SCK01}}$, SI01, SO01	$\overline{\text{SCK10}}$, SI10, SO10	$\overline{\text{SCK11}}$, SI11, SO11
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11
	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 to 16 bits			
Transfer rate	Max. f_{MCK}/6 [Hz][Notes 1, 2]			
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data input/output starts at the start of the operation of the serial clock. • DAPmn = 1: Data input/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 pins $\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$, and $\overline{\text{SCK11}}$ is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-65. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

(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	0	SOEm_1 0/1	SOEm_0 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	0	0	SSm_1 0/1	SSm_0 0/1

(d) 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	SCCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)

<R>

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

(f) Serial data register mn (SDRmn)

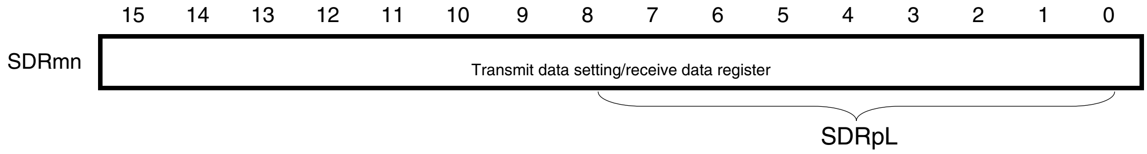
(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)
□ : 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

Figure 11-65. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (2/2)

(ii) During operation (SEm_n = 1) (lower 8 bits: SDRpL)



Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.

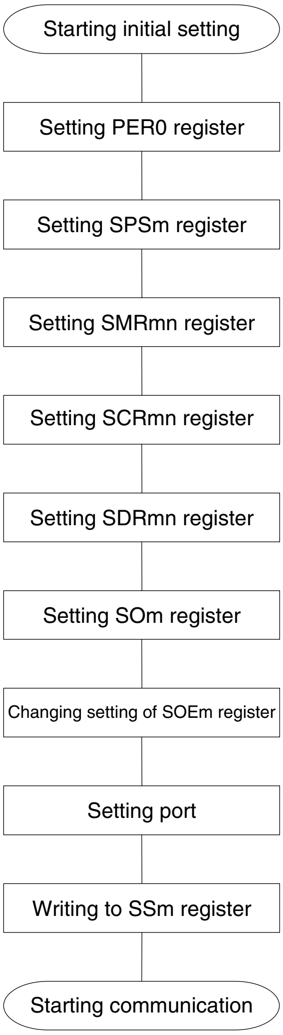
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

□ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-66. Initial Setting Procedure for Slave Transmission/Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

<R> Set bits 15 to 9 to any value for setting the baud rate.

Manipulate the SOmn bit and set an initial output level.

Set the SOEm_n bit to 1 and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

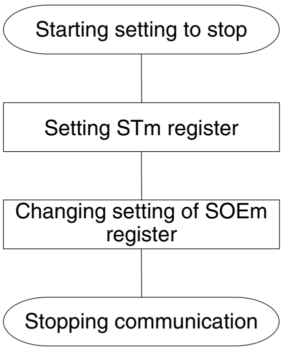
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Set transmit data to the SDRmn register and wait for a clock from the master.

Cautions 1. After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

2. Be sure to set transmit data to the SDRpL register before the clock from the master is started.

Figure 11-67. Procedure for Stopping Slave Transmission/Reception



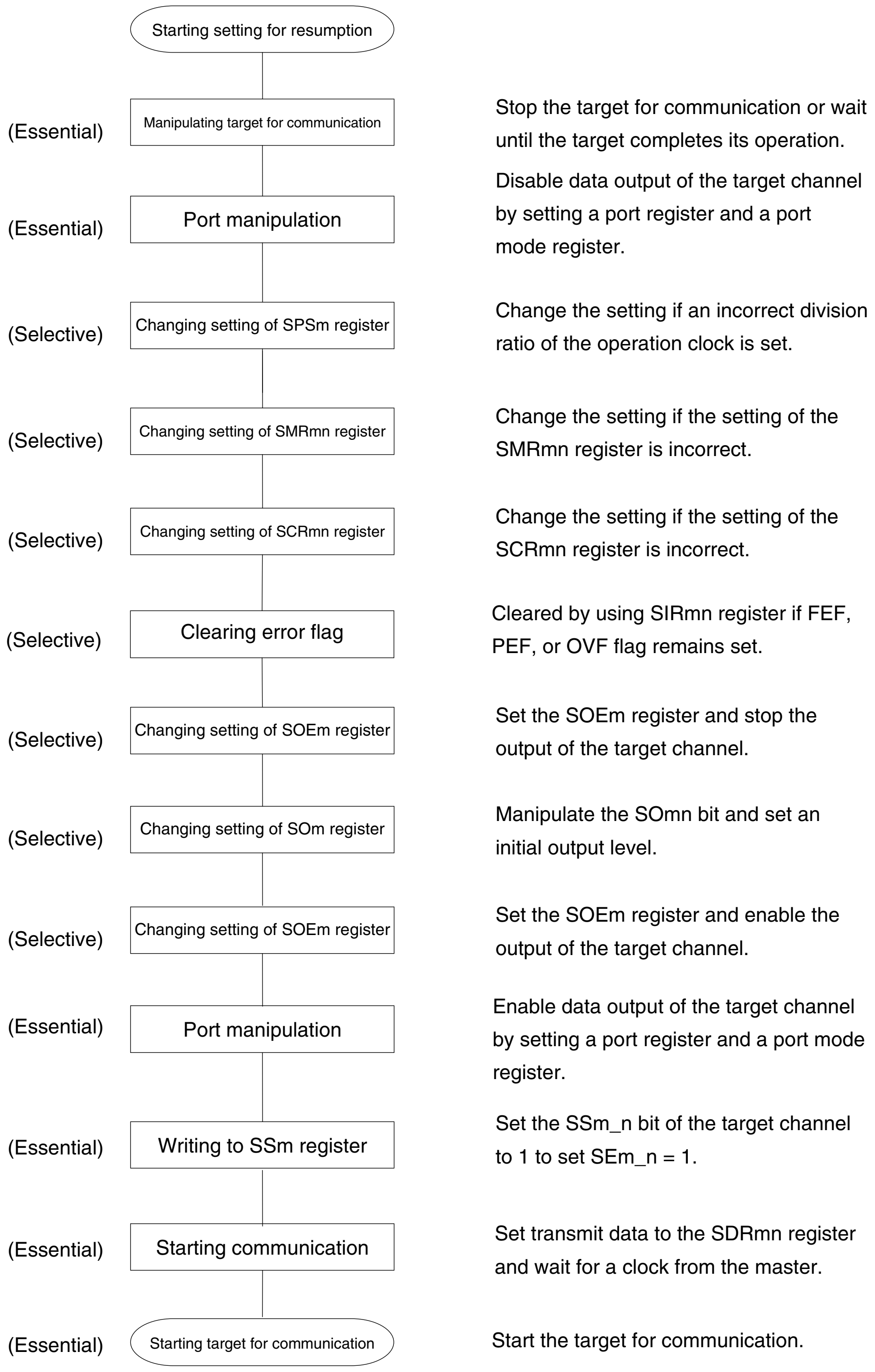
Write 1 to the STm_n bit of the target channel.

Clear the SOEm_n bit to 0 and stop the output of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-68 Procedure for Resuming Slave Transmission/Reception**).

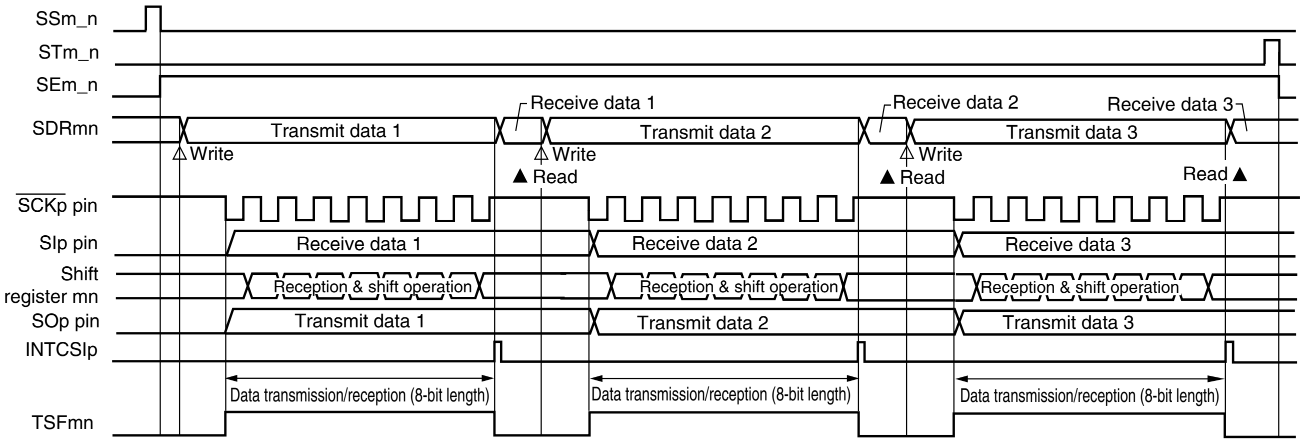
Figure 11-68. Procedure for Resuming Slave Transmission/Reception


Starting setting for resumption
(Essential)
Manipulating target for communication
Stop the target for communication or wait until the target completes its operation.
(Essential)
Port manipulation
Disable data output of the target channel by setting a port register and a port mode register.
(Selective)
Changing setting of SPSm register
Change the setting if an incorrect division ratio of the operation clock is set.
(Selective)
Changing setting of SMRmn register
Change the setting if the setting of the SMRmn register is incorrect.
(Selective)
Changing setting of SCRmn register
Change the setting if the setting of the SCRmn register is incorrect.
(Selective)
Clearing error flag
Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.
(Selective)
Changing setting of SOEm register
Set the SOEm register and stop the output of the target channel.
(Selective)
Changing setting of SOm register
Manipulate the SOmn bit and set an initial output level.
(Selective)
Changing setting of SOEm register
Set the SOEm register and enable the output of the target channel.
(Essential)
Port manipulation
Enable data output of the target channel by setting a port register and a port mode register.
(Essential)
Writing to SSm register
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.
(Essential)
Starting communication
Set transmit data to the SDRmn register and wait for a clock from the master.
(Essential)
Starting target for communication
Start the target for communication.

Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.

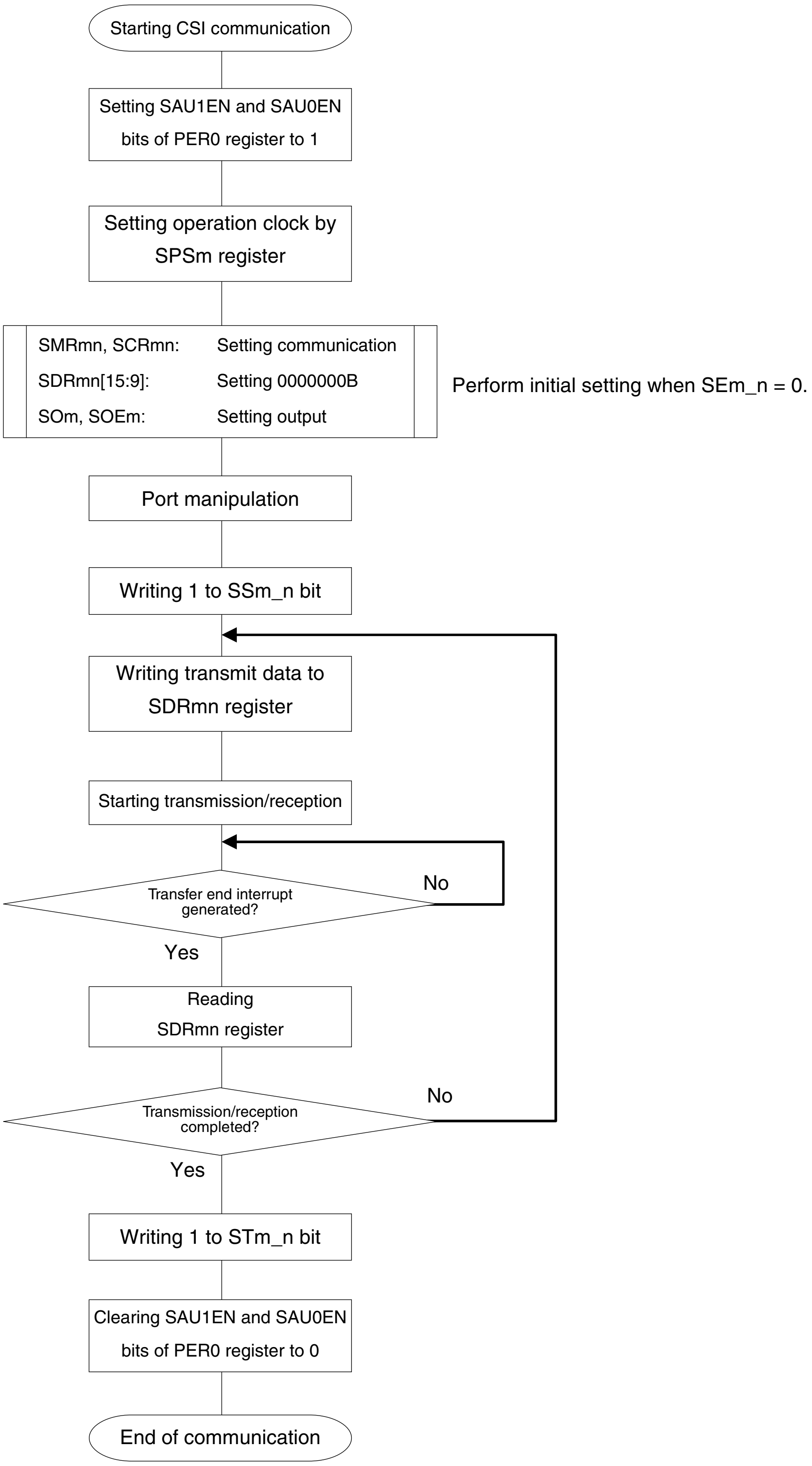
(3) Processing flow (in single-transmission/reception mode)

Figure 11-69. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 11-70. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)

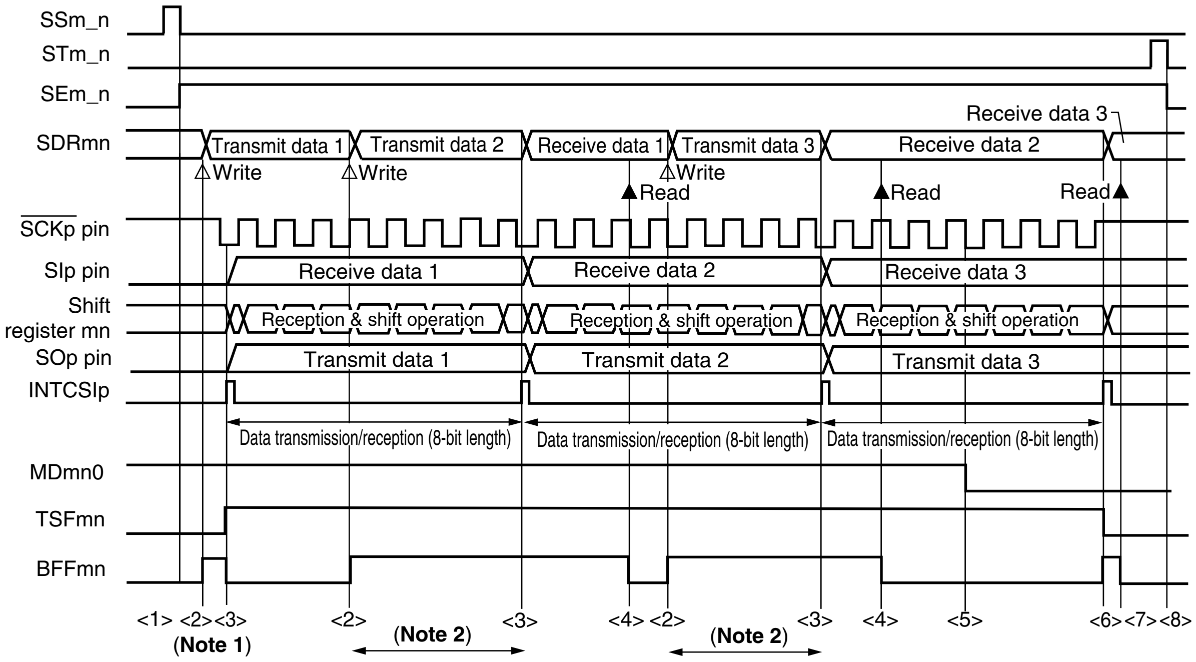

Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting 0000000B
SOm, SOEm: Setting output
Perform initial setting when SEm_n = 0.
Port manipulation
Writing 1 to SSm_n bit
Writing transmit data to SDRmn register
Starting transmission/reception
Transfer end interrupt generated?
No
Yes
Reading SDRmn register
Transmission/reception completed?
No
Yes
Writing 1 to STm_n bit
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

Cautions 1. After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

2. Be sure to set transmit data to the SDRpL register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-71. 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 11-72 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

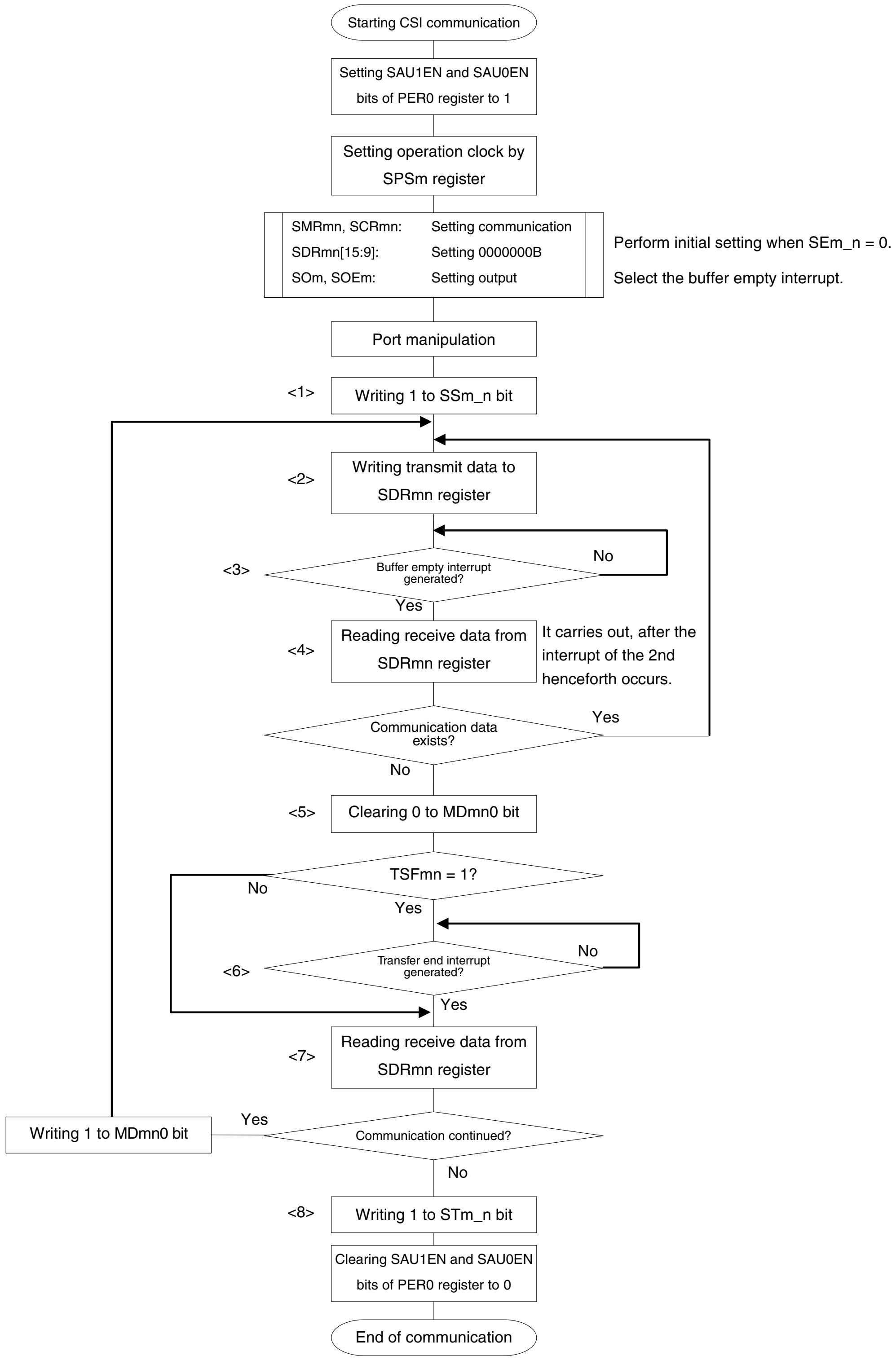
Figure 11-72. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)

<R>

<R>

<R>

<R>


Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting 0000000B
SOm, SOEm: Setting output
Perform initial setting when SEm_n = 0.
Select the buffer empty interrupt.
Port manipulation
<1>
Writing 1 to SSm_n bit
<2>
Writing transmit data to SDRmn register
<3>
Buffer empty interrupt generated?
No
Yes
<4>
Reading receive data from SDRmn register
It carries out, after the interrupt of the 2nd henceforth occurs.
Communication data exists?
Yes
No
<5>
Clearing 0 to MDmn0 bit
TSFmn = 1?
No
Yes
<6>
Transfer end interrupt generated?
No
Yes
<7>
Reading receive data from SDRmn register
Writing 1 to MDmn0 bit
Yes
Communication continued?
No
<8>
Writing 1 to STm_n bit
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

Cautions 1. After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

2. Be sure to set transmit data to the SDRpL register before the clock from the master is started.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-71 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

11.5.7 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) [Hz] = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

(2) Slave

(Transfer clock frequency) [Hz] = {Frequency of serial clock (f_{SCK}) supplied by master}[Note]

Note The permissible maximum transfer clock frequency is $f_{MCK}/6$.

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 = 0, 1), n: Channel number (n = 0, 1)

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11-4. Operating Clock Selection

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	24 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	12 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^{2}$	6 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^{3}$	3 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^{4}$	1.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^{5}$	750 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^{6}$	375 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^{7}$	187.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^{8}$	93.75 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^{9}$	46.86 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	23.44 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	11.72 kHz
	X	X	X	X	1	1	1	1	INTTM23	
1	0	0	0	0	X	X	X	X	f_{CLK}	24 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	12 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^{2}$	6 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^{3}$	3 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^{4}$	1.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^{5}$	750 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^{6}$	375 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^{7}$	187.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^{8}$	93.75 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^{9}$	46.86 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	23.44 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	11.72 kHz
	1	1	1	1	X	X	X	X	INTTM23	
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (STm = 0003H) the operation of the serial array unit m (SAUm). When selecting INTTM23 for the operation clock, also stop the timer array unit 2 (TAU2) (TT2 = 00FFH).

Remarks 1. X: Don’t care

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

11.5.8 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11) communication

The procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11) communication is described in Figure 11-73.

Figure 11-73. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data SDRmn register.	The 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.	Only error generated at the point of reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

11.6 Operation of SPI Function (CSI00, CSI01)

Channels 0 and 1 of SAU0 correspond to the SPI functions.

[Data transmission/reception]

- Data length of 7 to 16 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

[Expansion function]

- Slave select function of the SPI function

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^{2}C
0	0	CSI00 (supports SPI)	–	–
	1	CSI01 (supports SPI)		–
1	0	CSI10	–	–
	1	CSI11		IIC11
2	0	–	UART2	IIC20
	1	–		–

SPI function performs the following six types of communication operations.

• Master transmission	(See **11.6.1**.)
• Master reception	(See **11.6.2**.)
• Master transmission/reception	(See **11.6.3**.)
• Slave transmission	(See **11.6.4**.)
• Slave reception	(See **11.6.5**.)
• Slave transmission/reception	(See **11.6.6**.)

Multiple slaves can be connected to a master and communication can be performed by using the SPI function. The master outputs a slave select signal to the slave (one) that is the other party of communication, and each slave judges whether it has been selected as the other party of communication and controls the SO pin output. When a slave is selected, the SO pin enters an output state and transmit data can be communicated to the master. When a slave is not selected, the SO pin becomes high impedance to avoid shorting with the SO outputs of other slaves. Furthermore, when a slave is not selected, no transmission/reception operation is performed even if a serial clock is input from the master.

Caution Output the slave select signal by port manipulation.

Figure 11-74. Example of SPI Function Configuration

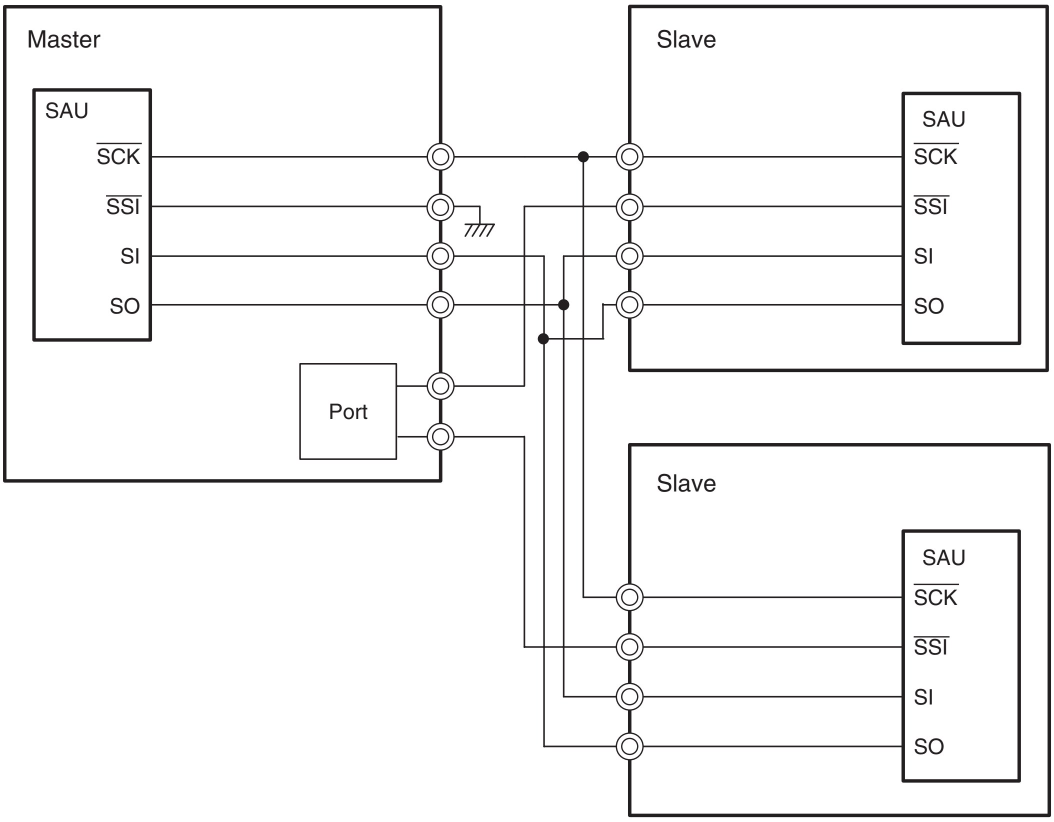
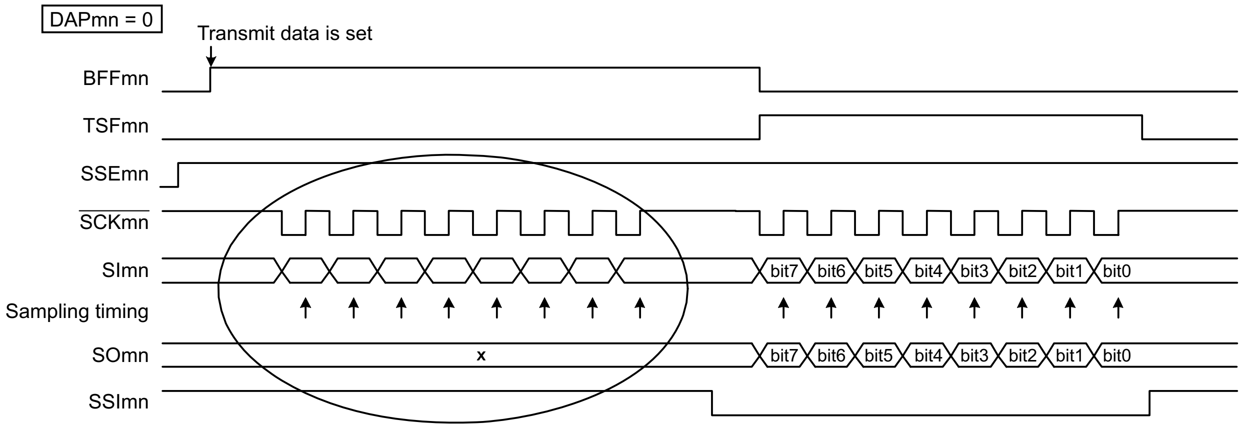
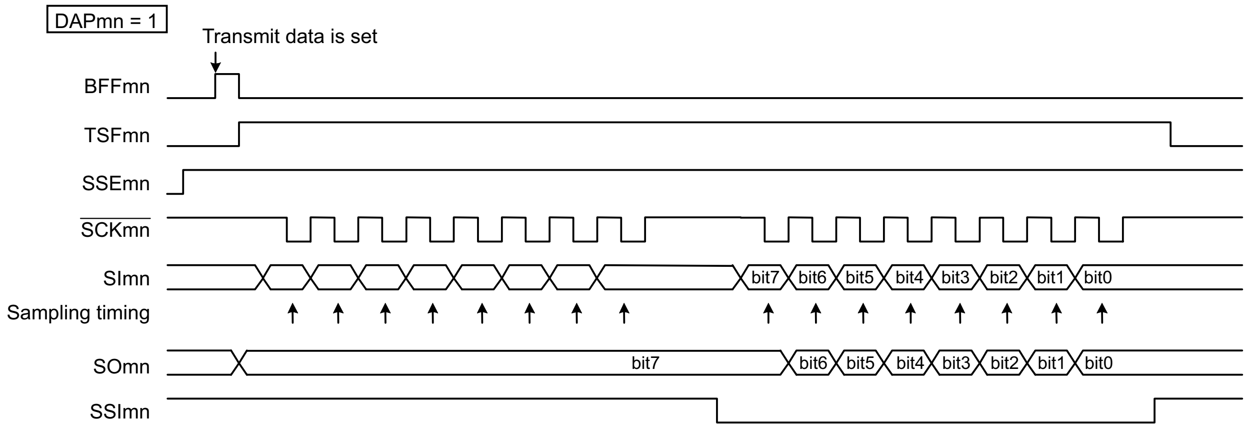


Figure 11-75. SPI Function Timing Diagram



While SSImn is at high level, transmission is not performed even if the falling edge of $\overline{\text{SCKmn}}$ (serial clock) arrives, and neither is receive data sampled in synchronization with the rising edge.
When SSImn goes to low level, data is output (shifted) in synchronization with the falling edge of the serial clock and a reception operation is performed in synchronization with the rising edge.

<R>



If DAPmn = 1, when transmit data is setshile SSImn is at high level, the first data (bit 7) is output to the data output. However, no shift operation is performed even if the rising edge of $\overline{\text{SCKmn}}$ (serial clock) arrives, and neither is receive data sampled in synchronization with the falling edge. WhenSSImn goes to low level, data is output (shifted) in synchronization with the next rising edge and a reception operation is performed in synchronization with the falling edge.

11.6.1 Master transmission

Master transmission is that the 78K0R/Hx3 outputs a transfer clock and transmits data to another device.

SPI Function	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	$\overline{\text{SCK00}}$, SO00	$\overline{\text{SCK01}}$, SO01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	7 to 16 bits	
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz]^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 29 ELECTRICAL SPECIFICATIONS**).

(1) Register setting

Figure 11-76. Example of Contents of Registers for Master Transmission of SPI Function (CSI00, CSI01) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

Communication starts when these bits are 1 if the data phase is forward (CKPmn = 0). If the phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(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	0	SOEm_1 0/1	SOEm_0 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	0	0	SSm_1 0/1	SSm_0 0/1

(d) 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	SCCSmn 0	0	0	0	0	0	STSmn 0	0	SISm0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)

<R>

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

(f) Serial data register mn (SDRmn)

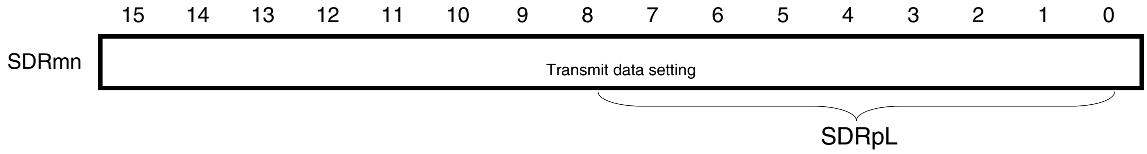
(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)
□: Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-76. Example of Contents of Registers for Master Transmission of SPI Function (CSI00, CSI01) (2/2)

(ii) During operation (SEm_n = 1) (lower 8 bits: SDRpL)



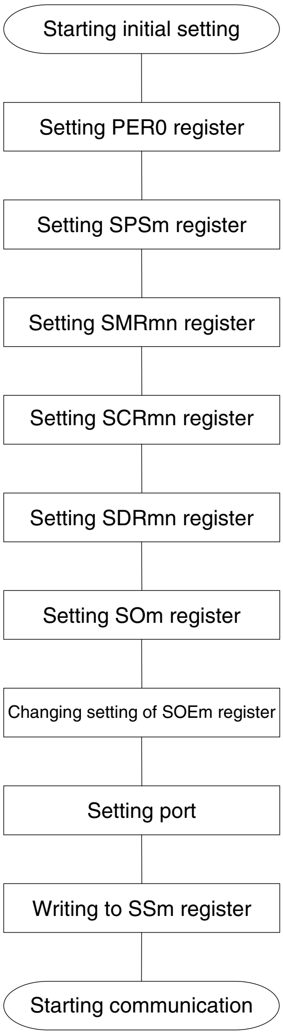
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

□ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-77. Initial Setting Procedure for Master Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate.

Manipulate the SOmn and CKOmn bits and set an initial output level.

Set the SOEm_n bit to 1 and enable data output of the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

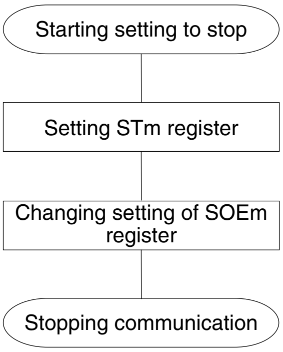
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Set transmit data to the SDRmn register and start communication.

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 11-78. Procedure for Stopping Master Transmission



Write 1 to the STm_n bit of the target channel.

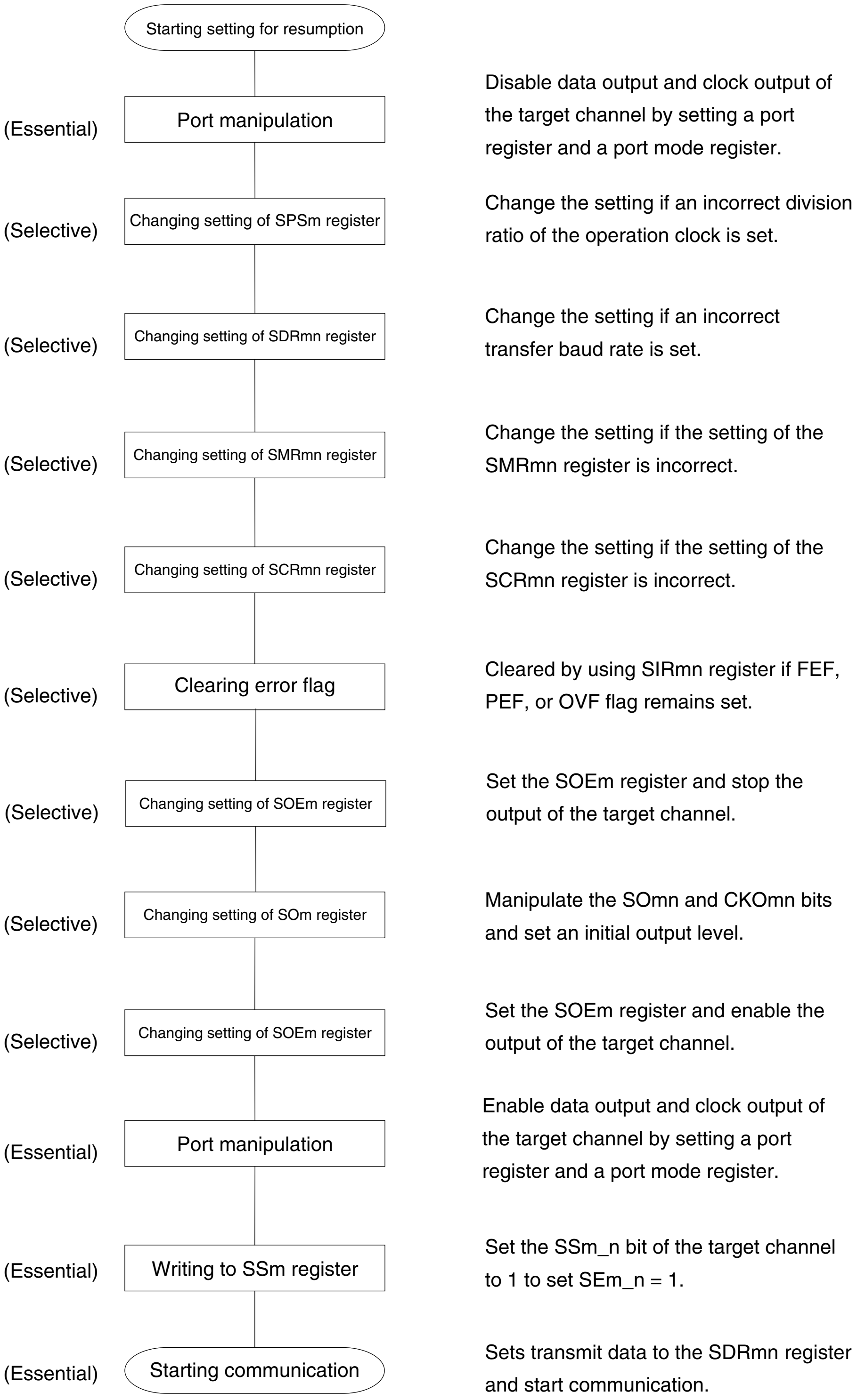
Clear the SOEm_n bit to 0 and stop the output of the target channel.

Stop communication in midway.

Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-79 Procedure for Resuming Master Transmission**).

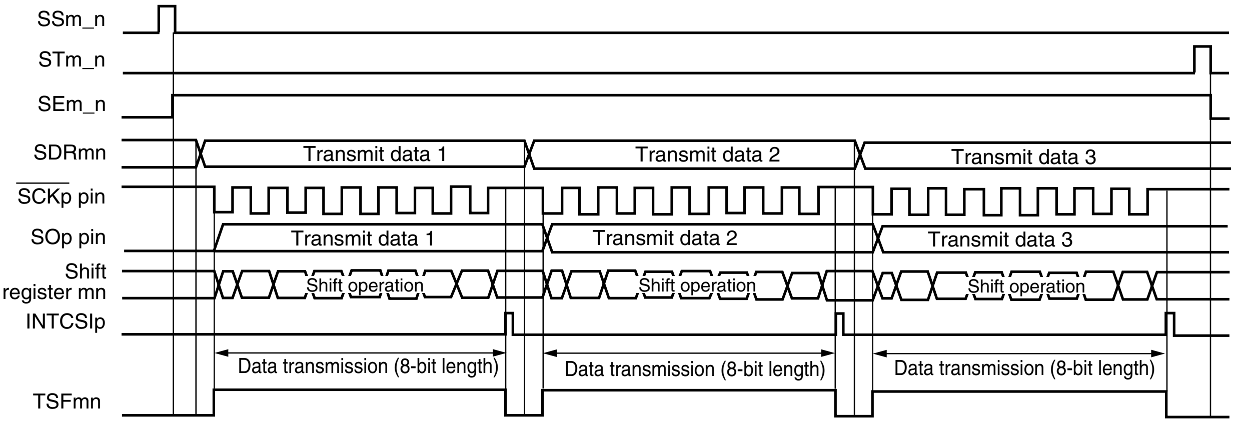
2. m: Unit number (m = 0) , n: Channel number (n = 0, 1)

Figure 11-79. Procedure for Resuming Master Transmission


Starting setting for resumption
(Essential)
Port manipulation
Disable data output and clock output of the target channel by setting a port register and a port mode register.
(Selective)
Changing setting of SPSm register
Change the setting if an incorrect division ratio of the operation clock is set.
(Selective)
Changing setting of SDRmn register
Change the setting if an incorrect transfer baud rate is set.
(Selective)
Changing setting of SMRmn register
Change the setting if the setting of the SMRmn register is incorrect.
(Selective)
Changing setting of SCRmn register
Change the setting if the setting of the SCRmn register is incorrect.
(Selective)
Clearing error flag
Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.
(Selective)
Changing setting of SOEm register
Set the SOEm register and stop the output of the target channel.
(Selective)
Changing setting of SOm register
Manipulate the SOmn and CKOmn bits and set an initial output level.
(Selective)
Changing setting of SOEm register
Set the SOEm register and enable the output of the target channel.
(Essential)
Port manipulation
Enable data output and clock output of the target channel by setting a port register and a port mode register.
(Essential)
Writing to SSm register
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.
(Essential)
Starting communication
Sets transmit data to the SDRmn register and start communication.

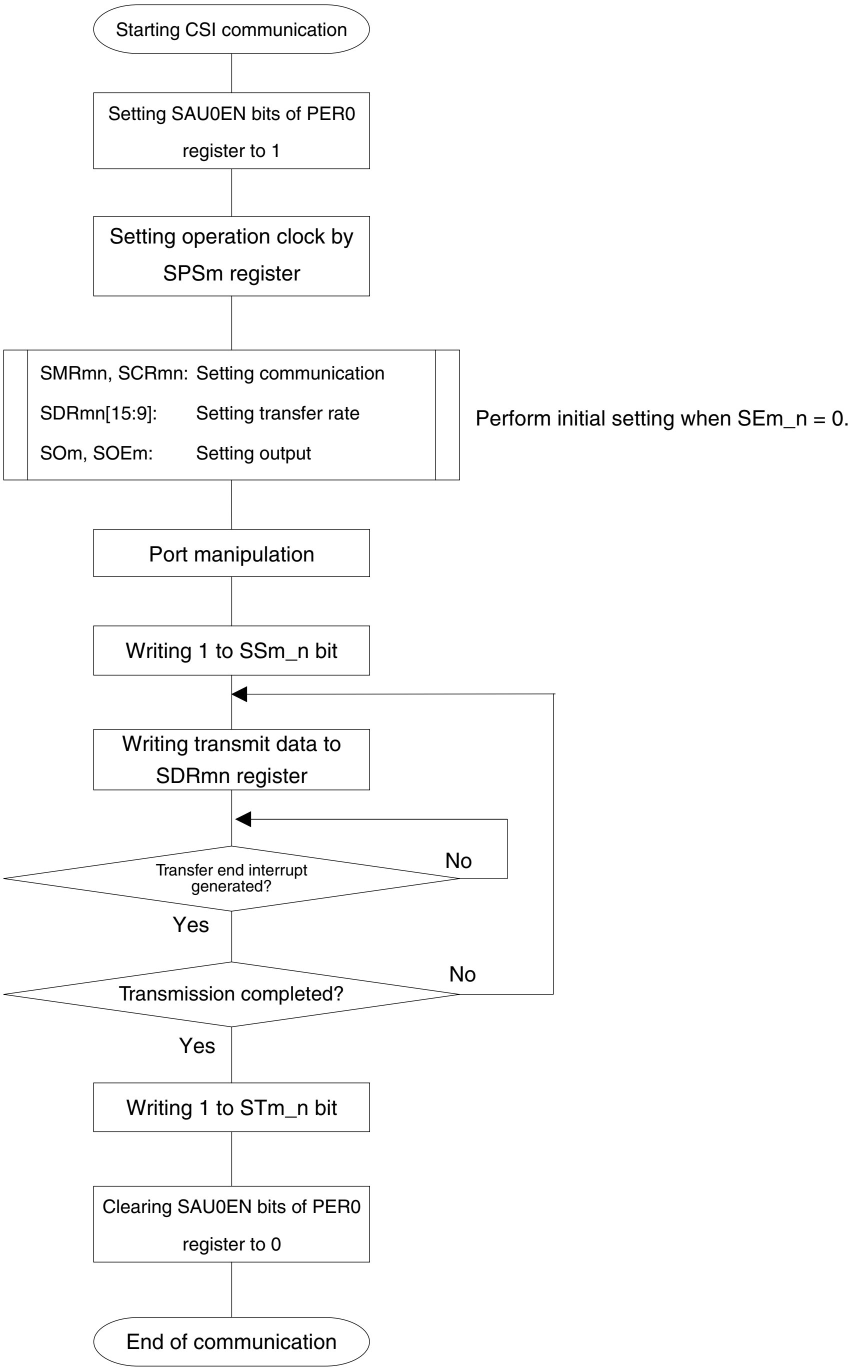
(3) Processing flow (in single-transmission mode)

Figure 11-80. Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

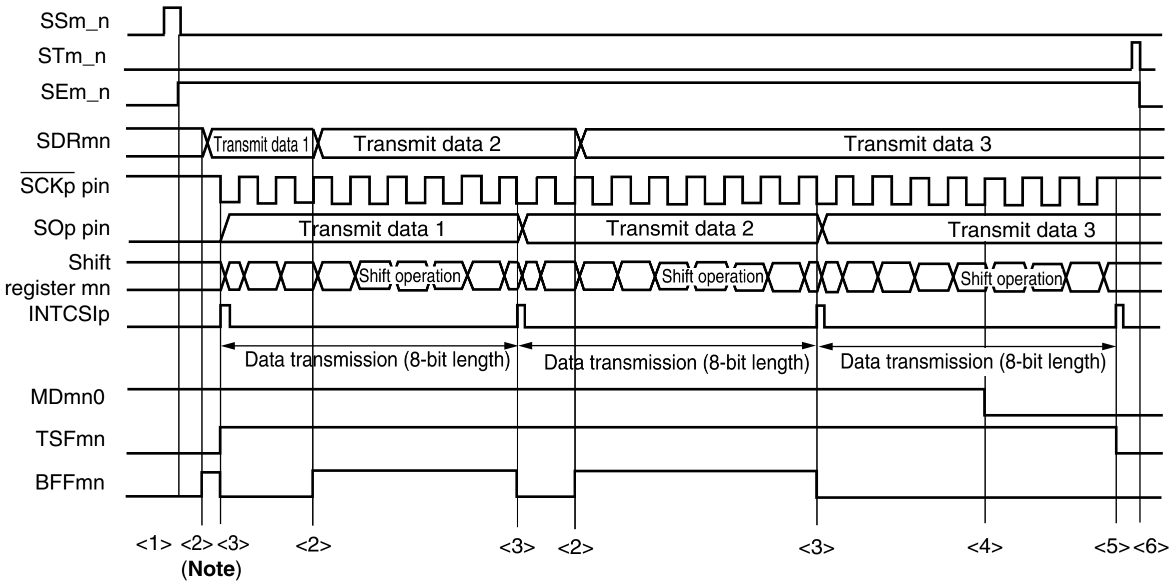
Figure 11-81. Flowchart of Master Transmission (in Single-Transmission Mode)


Starting CSI communication
Setting SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting transfer rate
SOm, SOEm: Setting output
Perform initial setting when SEm_n = 0.
Port manipulation
Writing 1 to SSm_n bit
Writing transmit data to SDRmn register
Transfer end interrupt generated?
No
Yes
Transmission completed?
No
Yes
Writing 1 to STm_n bit
Clearing SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 11-82. Timing Chart of Master 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.

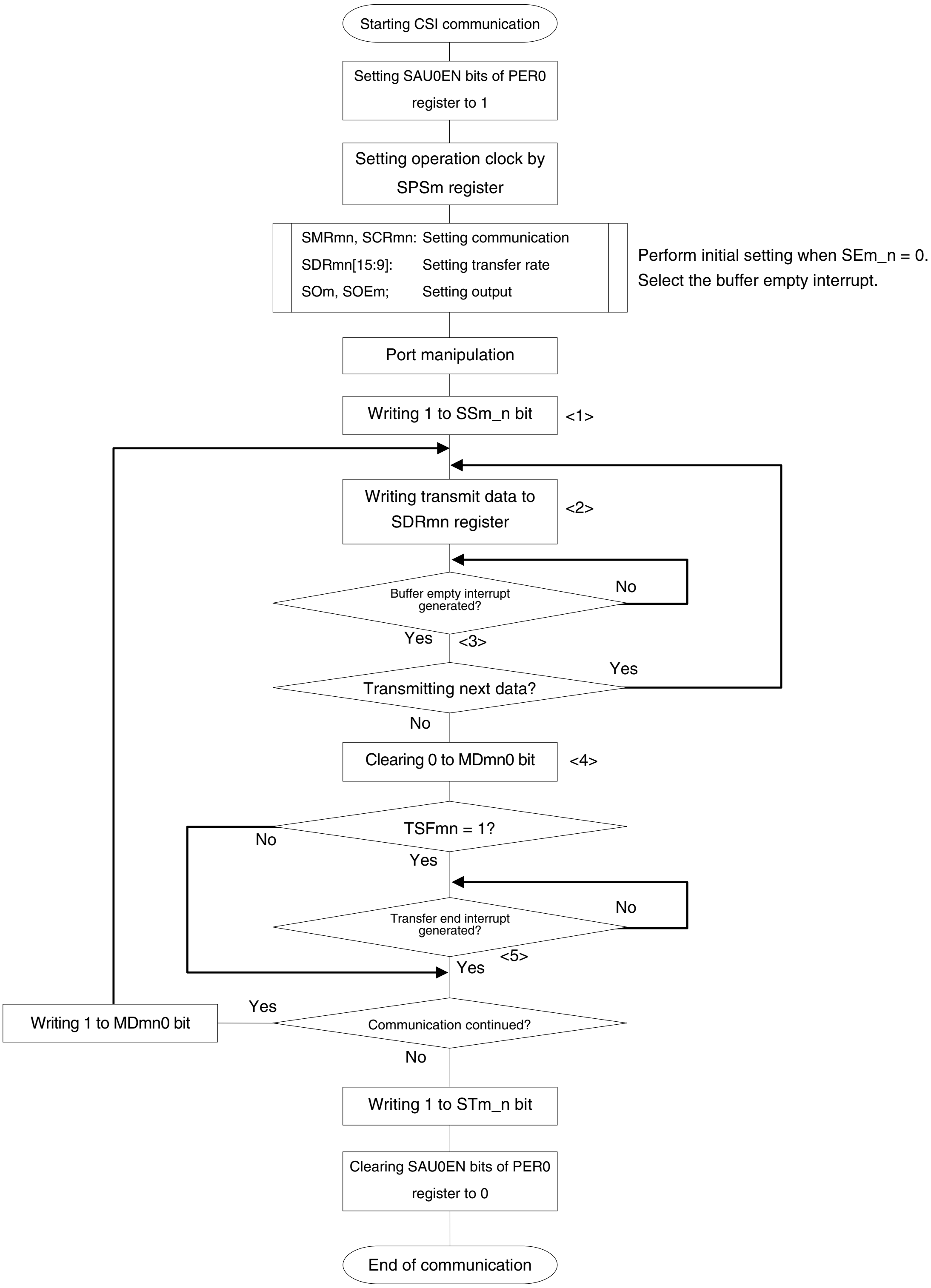
Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 11-83. Flowchart of Master Transmission (in Continuous Transmission Mode)

<R>

<R>


Starting CSI communication
Setting SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting transfer rate
SOm, SOEm; Setting output
Perform initial setting when SEm_n = 0.
Select the buffer empty interrupt.
Port manipulation
Writing 1 to SSm_n bit
<1>
Writing transmit data to SDRmn register
<2>
Buffer empty interrupt generated?
No
Yes
<3>
Transmitting next data?
Yes
No
Clearing 0 to MDmn0 bit
<4>
TSFmn = 1?
No
Yes
Transfer end interrupt generated?
No
<5>
Yes
Writing 1 to MDmn0 bit
Yes
Communication continued?
No
Writing 1 to STm_n bit
Clearing SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-82 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

11.6.2 Master reception

Master reception is that the 78K0R/Hx3 outputs a transfer clock and receives data from other device.

SPI Function	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	$\overline{\text{SCK00}}$, SI00	$\overline{\text{SCK01}}$, SI01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 to 16 bits	
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz][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 29 ELECTRICAL SPECIFICATIONS**).

(1) Register setting

Figure 11-84. Example of Contents of Registers for Master Reception of SPI Function (CSI00, CSI01) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 ×	SOm0 ×

Communication starts when these bits are 1 if the data phase is forward (CKPmn = 0). If the phase is reversed (CKPmn = 1), communication starts when these bits are 0.

<R> **(b) Serial output enable register m (SOEm) …Set the receive target channel to 0.**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm_1 0/1	SOEm_0 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	0	0	SSm_1 0/1	SSm_0 0/1

(d) 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	SCCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<R> SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

(f) Serial data register mn (SDRmn)

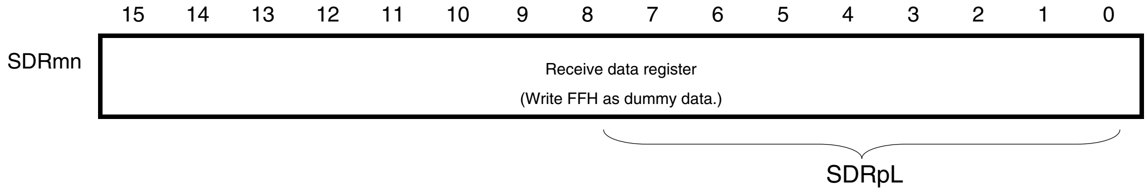
(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)
□ : 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

Figure 11-84. Example of Contents of Registers for Master Reception of SPI Function (CSI00, CSI01) (2/2)

(ii) During operation (SEm_n = 1) (lower 8 bits: SDRpL)



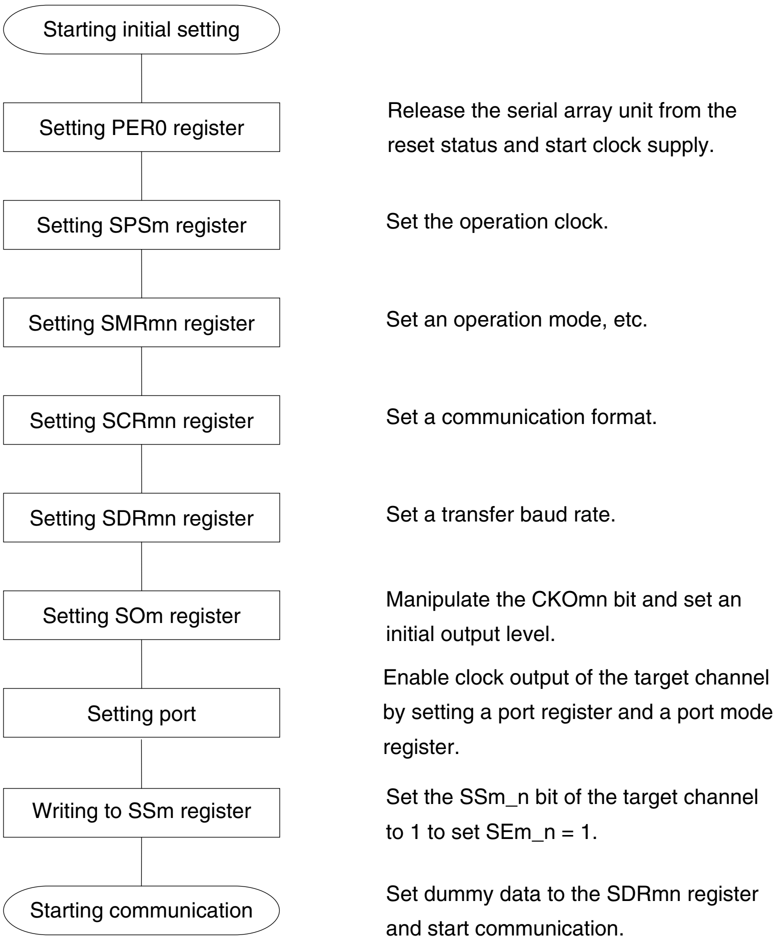
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

□: Setting is fixed in the CSI master transmission mode, ▨: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

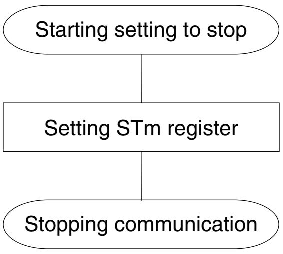
(2) Operation procedure

Figure 11-85. Initial Setting Procedure for Master Reception


Starting initial setting
Setting PER0 register
Release the serial array unit from the reset status and start clock supply.
Setting SPSm register
Set the operation clock.
Setting SMRmn register
Set an operation mode, etc.
Setting SCRmn register
Set a communication format.
Setting SDRmn register
Set a transfer baud rate.
Setting SOm register
Manipulate the CKOmn bit and set an initial output level.
Setting port
Enable clock output of the target channel by setting a port register and a port mode register.
Writing to SSm register
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.
Starting communication
Set dummy data to the SDRmn register and start communication.

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-86. Procedure for Stopping Master Reception

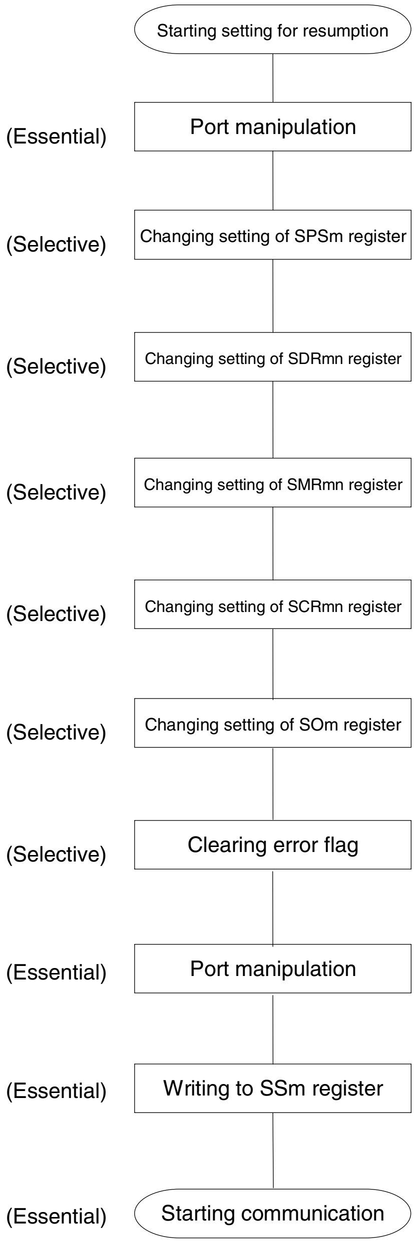


Write 1 to the STm_n bit of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-87 Procedure for Resuming Master Reception**).

Figure 11-87. Procedure for Resuming Master Reception



Disable clock output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if an incorrect transfer baud rate is set.

Change the setting if the setting of the SMRmn register is incorrect.

Change the setting if the setting of the SCRmn register is incorrect.

Manipulate the CKOmn bit and set a clock output level.

Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.

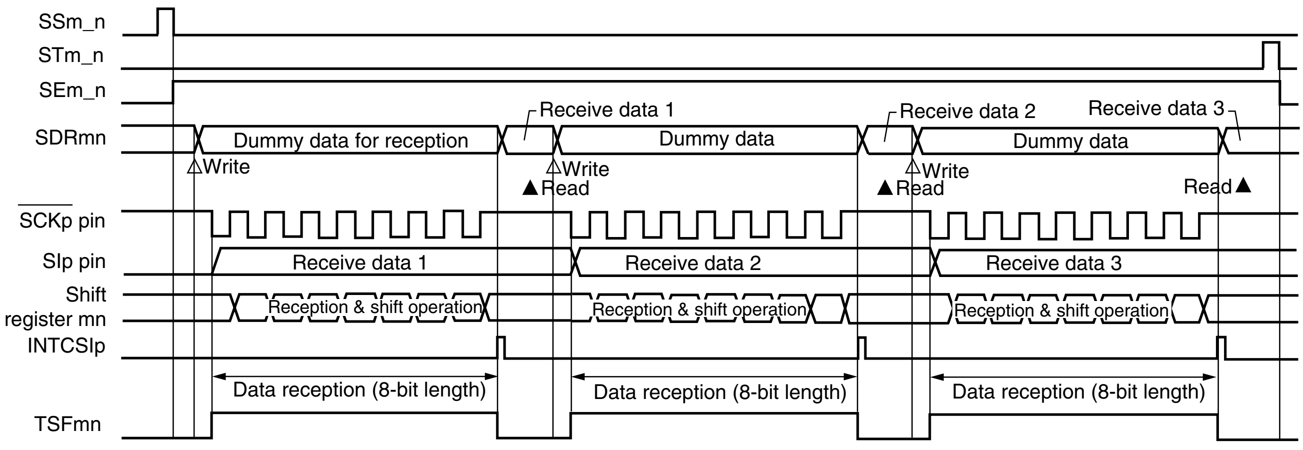
Enable clock output of the target channel by setting a port register and a port mode register.

Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Set dummy data to the SDRmn register and start communication.

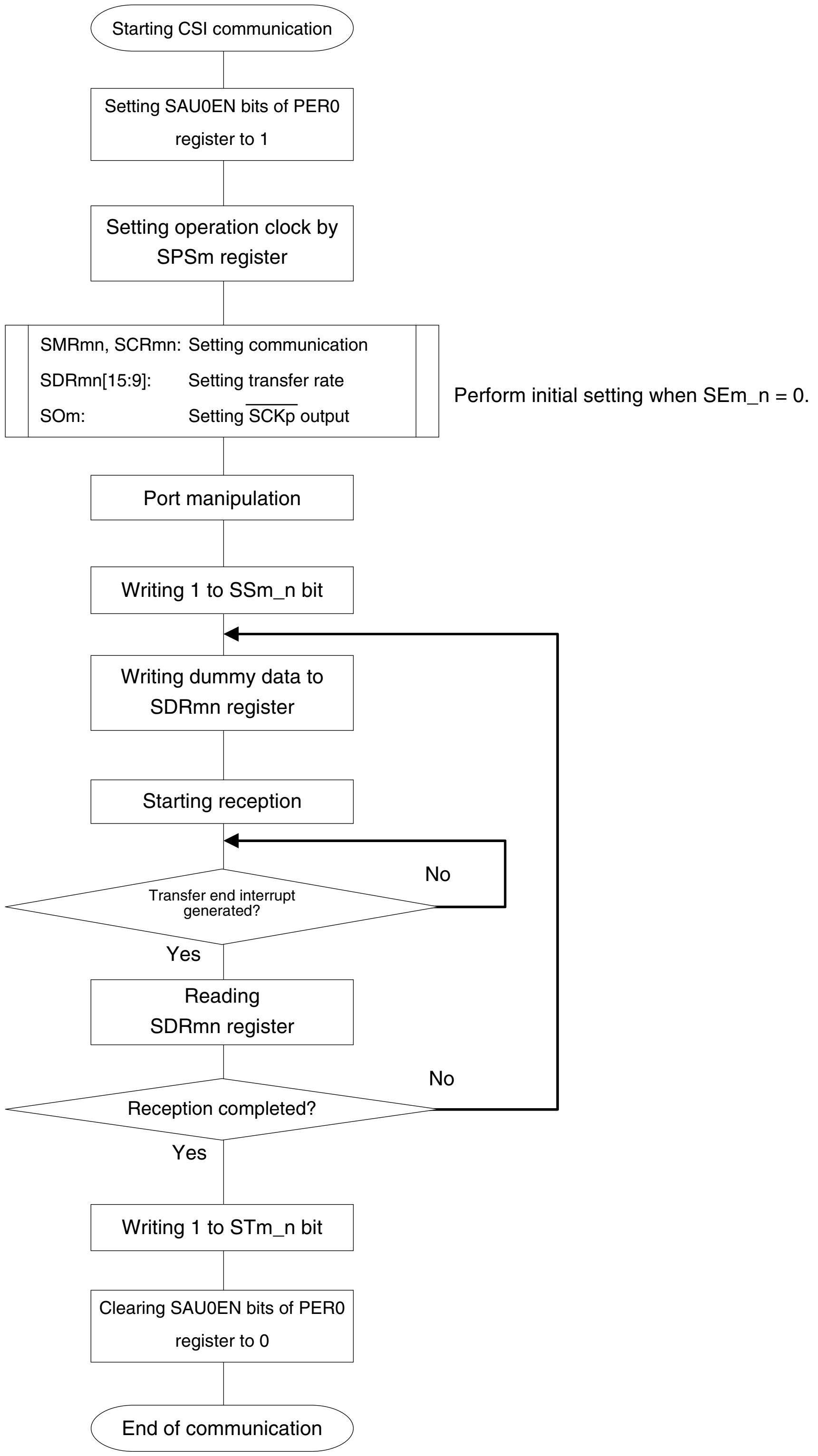
(3) Processing flow (in single-reception mode)

Figure 11-88. Timing Chart of Master Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

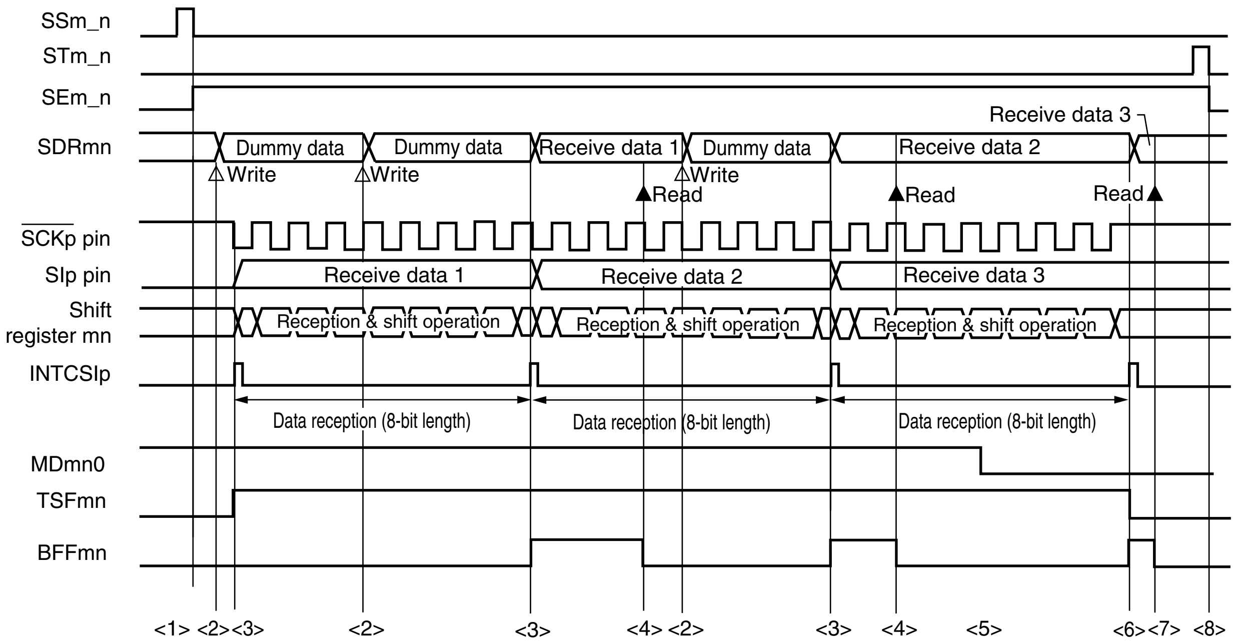
Figure 11-89. Flowchart of Master Reception (in Single-Reception Mode)


Starting CSI communication
Setting SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting transfer rate
SOm: Setting $\overline{\text{SCKp}}$ output
Perform initial setting when SEm_n = 0.
Port manipulation
Writing 1 to SSm_n bit
Writing dummy data to SDRmn register
Starting reception
Transfer end interrupt generated?
No
Yes
Reading SDRmn register
Reception completed?
No
Yes
Writing 1 to STm_n bit
Clearing SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous reception mode)

Figure 11-90. Timing Chart of Master Reception (in Continuous Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-91 Flowchart of Master Reception (in Continuous Reception Mode)**.

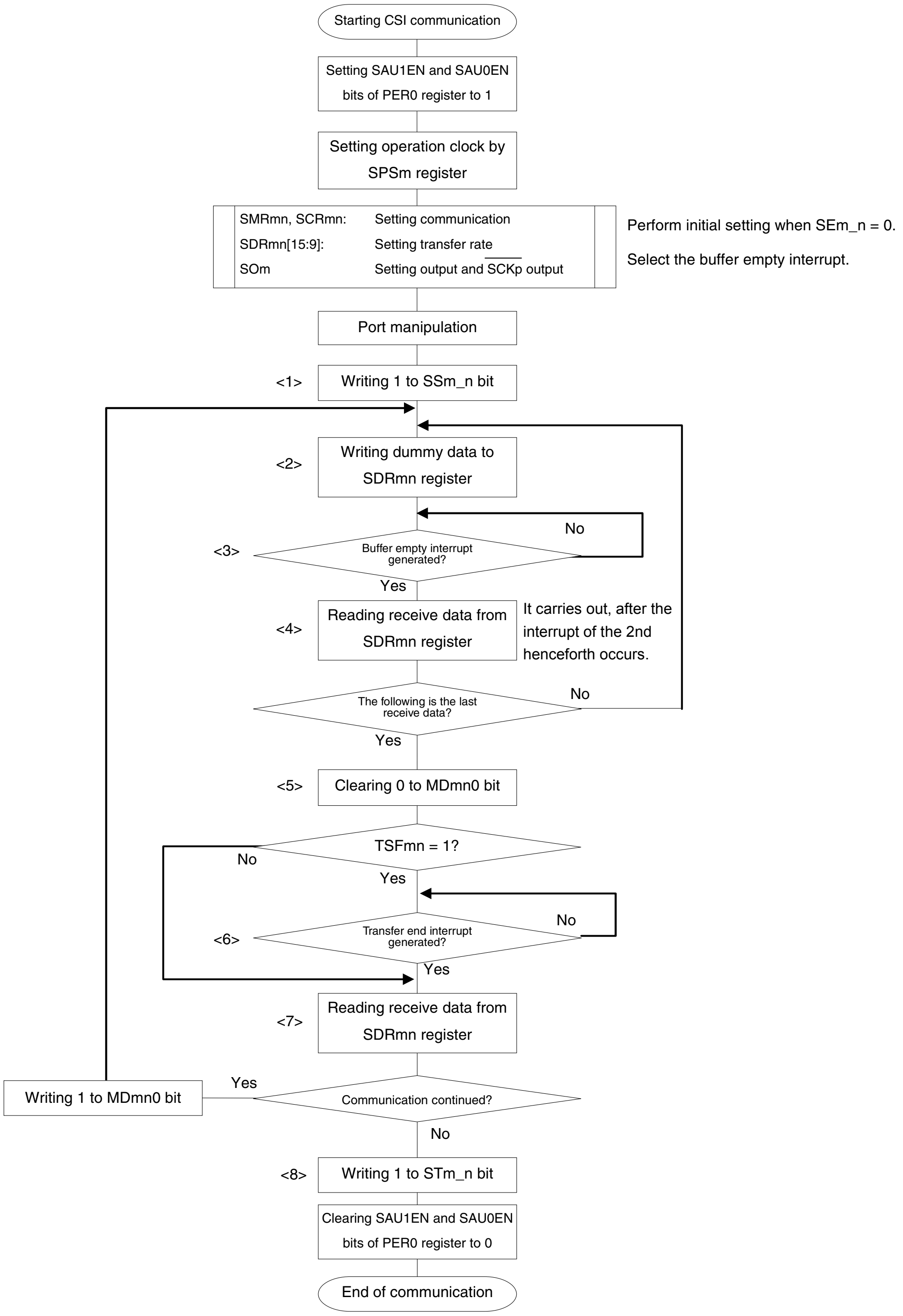
2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 11-91. Flowchart of Master Reception (in Continuous Reception Mode)

<R>

<R>

<R>


Starting CSI communication
Setting SAU1EN and SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting transfer rate
SOm Setting output and SCKp output
Perform initial setting when SEm_n = 0.
Select the buffer empty interrupt.
Port manipulation
<1> Writing 1 to SSm_n bit
<2> Writing dummy data to SDRmn register
<3> Buffer empty interrupt generated?
No
Yes
<4> Reading receive data from SDRmn register
It carries out, after the interrupt of the 2nd henceforth occurs.
The following is the last receive data?
No
Yes
<5> Clearing 0 to MDmn0 bit
TSFmn = 1?
No
Yes
<6> Transfer end interrupt generated?
No
Yes
<7> Reading receive data from SDRmn register
Communication continued?
Yes
Writing 1 to MDmn0 bit
No
<8> Writing 1 to STm_n bit
Clearing SAU1EN and SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-90 Timing Chart of Master Reception (in Continuous Reception Mode)**.

11.6.3 Master transmission/reception

Master transmission/reception is that the 78K0R/Hx3 outputs a transfer clock and transmits/receives data to/from other device.

SPI Function	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	$\overline{\text{SCK00}}$, SI00, SO00	$\overline{\text{SCK01}}$, SI01, SO01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 to 16 bits	
Transfer rate	Max. f_{CLK}/4 [Hz], Min. f_{CLK}/(2 × 2^{11} × 128) [Hz][Note] f_{CLK}: System clock frequency	
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data input/output starts at the start of the operation of the serial clock. • DAPmn = 1: Data input/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 29 ELECTRICAL SPECIFICATIONS**).

(1) Register setting

Figure 11-92. Example of Contents of Registers for Master Transmission/Reception of SPI Function (CSI00, CSI01) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	0/1	0/1	0	0	0	0	0	0	0/1	0/1

Communication starts when these bits are 1 if the data phase is forward (CKPmn = 0). If the phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(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															SOEm_1	SOEm_0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	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															SSm_1	SSm_0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	SCCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)

<R>

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn			PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0	DLSmn3	DLSmn2	DLSmn1	DLSmn0
	1	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0/1	0/1	0/1	0/1

(f) Serial data register mn (SDRmn)

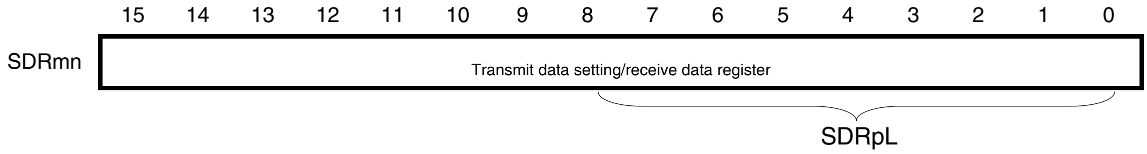
(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)
□ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-92. Example of Contents of Registers for Master Transmission/Reception of SPI Function (CSI00, CSI01) (2/2)

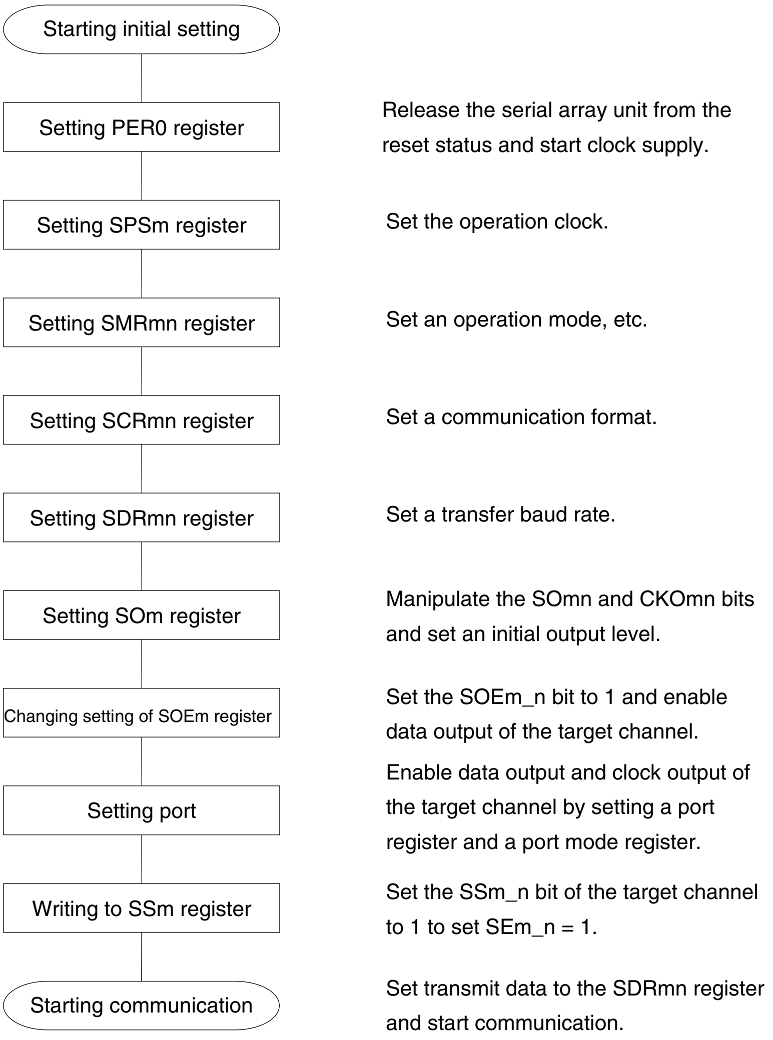
(ii) During operation (SEm_n = 1) (lower 8 bits: SDRpL)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)
□: Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

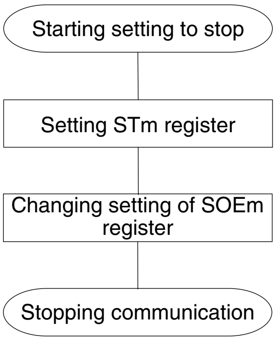
(2) Operation procedure

Figure 11-93. Initial Setting Procedure for Master Transmission/Reception


Starting initial setting
Setting PER0 register
Release the serial array unit from the reset status and start clock supply.
Setting SPSm register
Set the operation clock.
Setting SMRmn register
Set an operation mode, etc.
Setting SCRmn register
Set a communication format.
Setting SDRmn register
Set a transfer baud rate.
Setting SOm register
Manipulate the SOmn and CKOmn bits and set an initial output level.
Changing setting of SOEm register
Set the SOEm_n bit to 1 and enable data output of the target channel.
Setting port
Enable data output and clock output of the target channel by setting a port register and a port mode register.
Writing to SSm register
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.
Starting communication
Set transmit data to the SDRmn register and start communication.

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-94. Procedure for Stopping Master Transmission/Reception



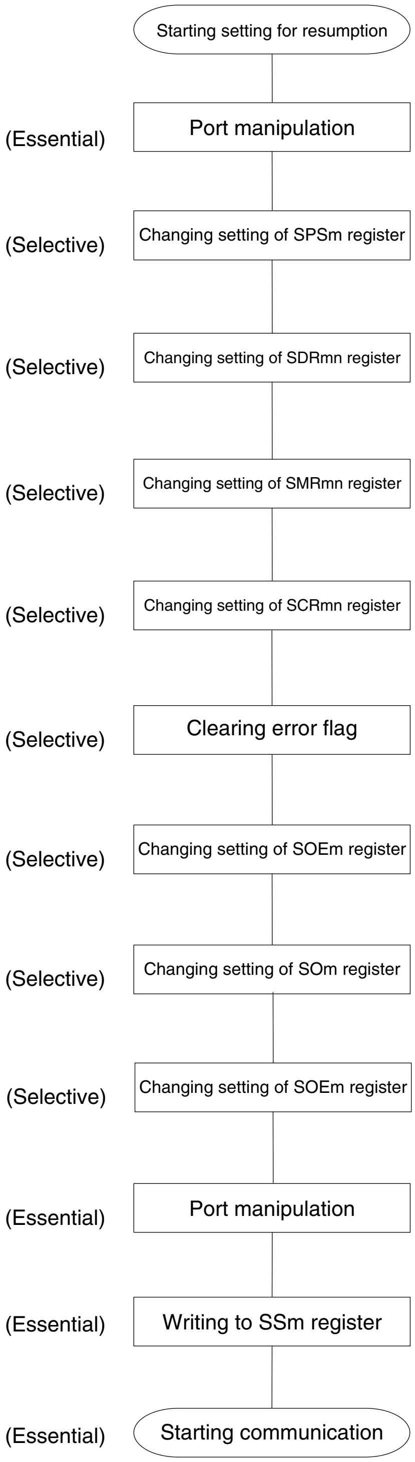
Write 1 to the STm_n bit of the target channel.

Clear the SOEm_n bit to 0 and stop the output of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-95 Procedure for Resuming Master Transmission/Reception**).

Figure 11-95. Procedure for Resuming Master Transmission/Reception


Starting setting for resumption
(Essential)
Port manipulation
(Selective)
Changing setting of SPSm register
(Selective)
Changing setting of SDRmn register
(Selective)
Changing setting of SMRmn register
(Selective)
Changing setting of SCRmn register
(Selective)
Clearing error flag
(Selective)
Changing setting of SOEm register
(Selective)
Changing setting of SOm register
(Selective)
Changing setting of SOEm register
(Essential)
Port manipulation
(Essential)
Writing to SSm register
(Essential)
Starting communication

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if an incorrect transfer baud rate is set.

Change the setting if the setting of the SMRmn register is incorrect.

Change the setting if the setting of the SCRmn register is incorrect.

Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.

Set the SOEm register and stop the output of the target channel.

Manipulate the SOmn and CKOmn bits and set an initial output level.

Set the SOEm register and enable the output of the target channel.

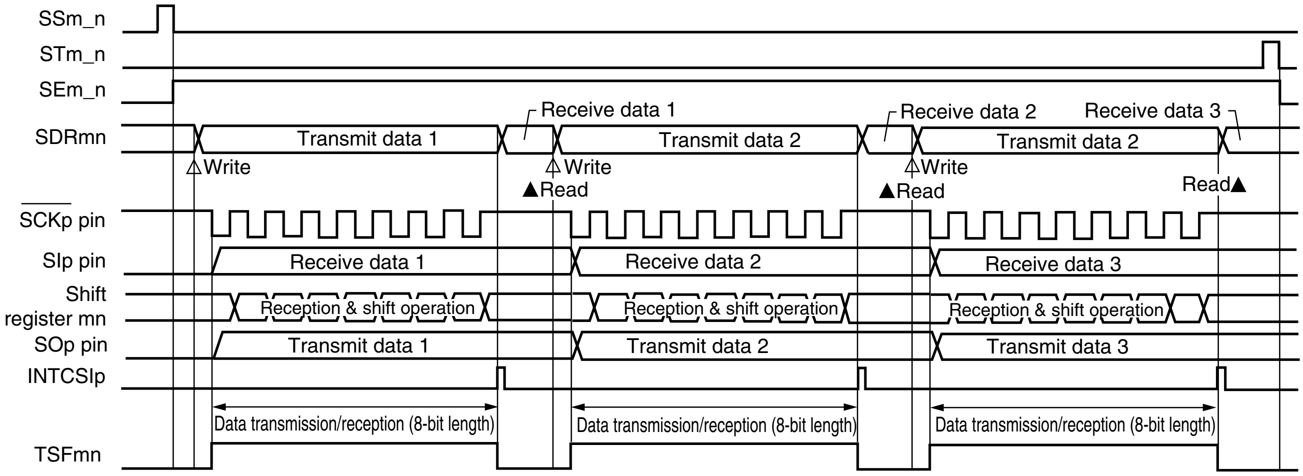
Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Sets transmit data to the SDRmn register and start communication.

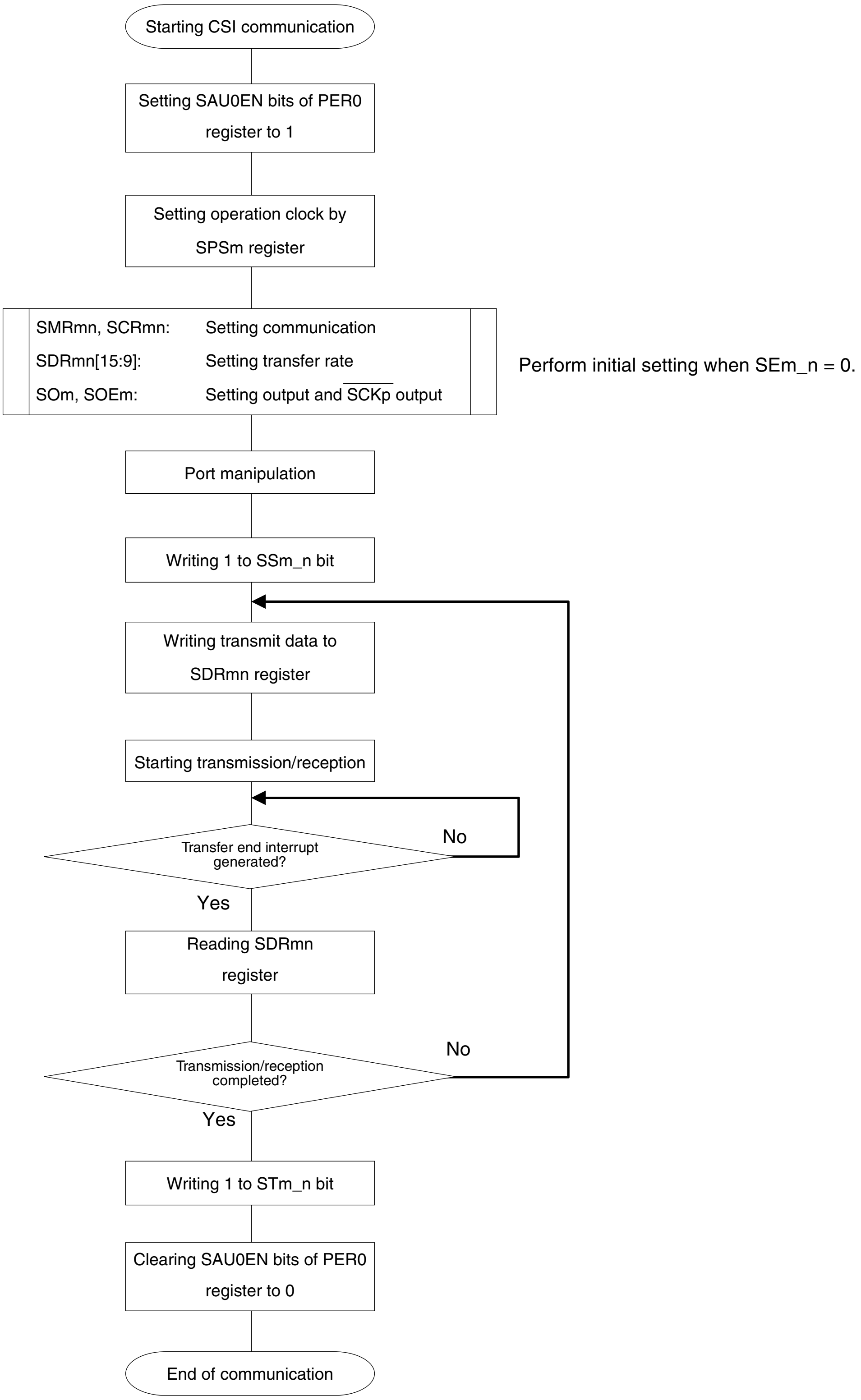
(3) Processing flow (in single-transmission/reception mode)

Figure 11-96. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

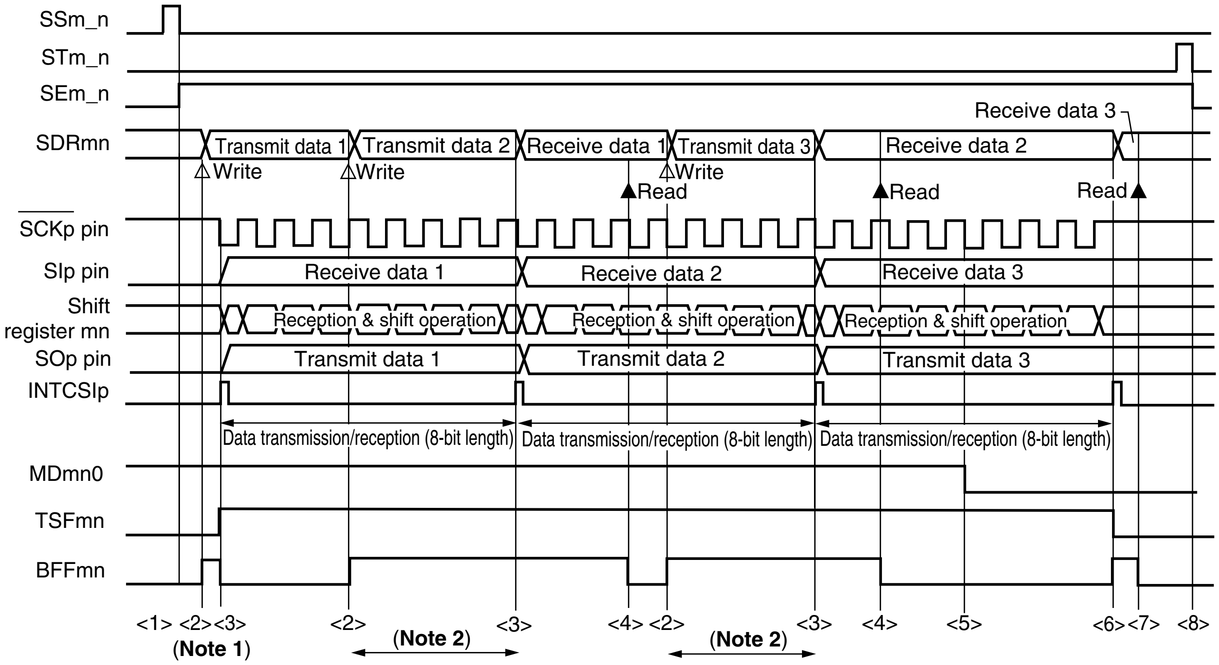
Figure 11-97. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)


Starting CSI communication
Setting SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting transfer rate
SOm, SOEm: Setting output and $\overline{\text{SCKp}}$ output
Perform initial setting when SEm_n = 0.
Port manipulation
Writing 1 to SSm_n bit
Writing transmit data to SDRmn register
Starting transmission/reception
Transfer end interrupt generated?
No
Yes
Reading SDRmn register
Transmission/reception completed?
No
Yes
Writing 1 to STm_n bit
Clearing SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-98. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



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 11-99 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

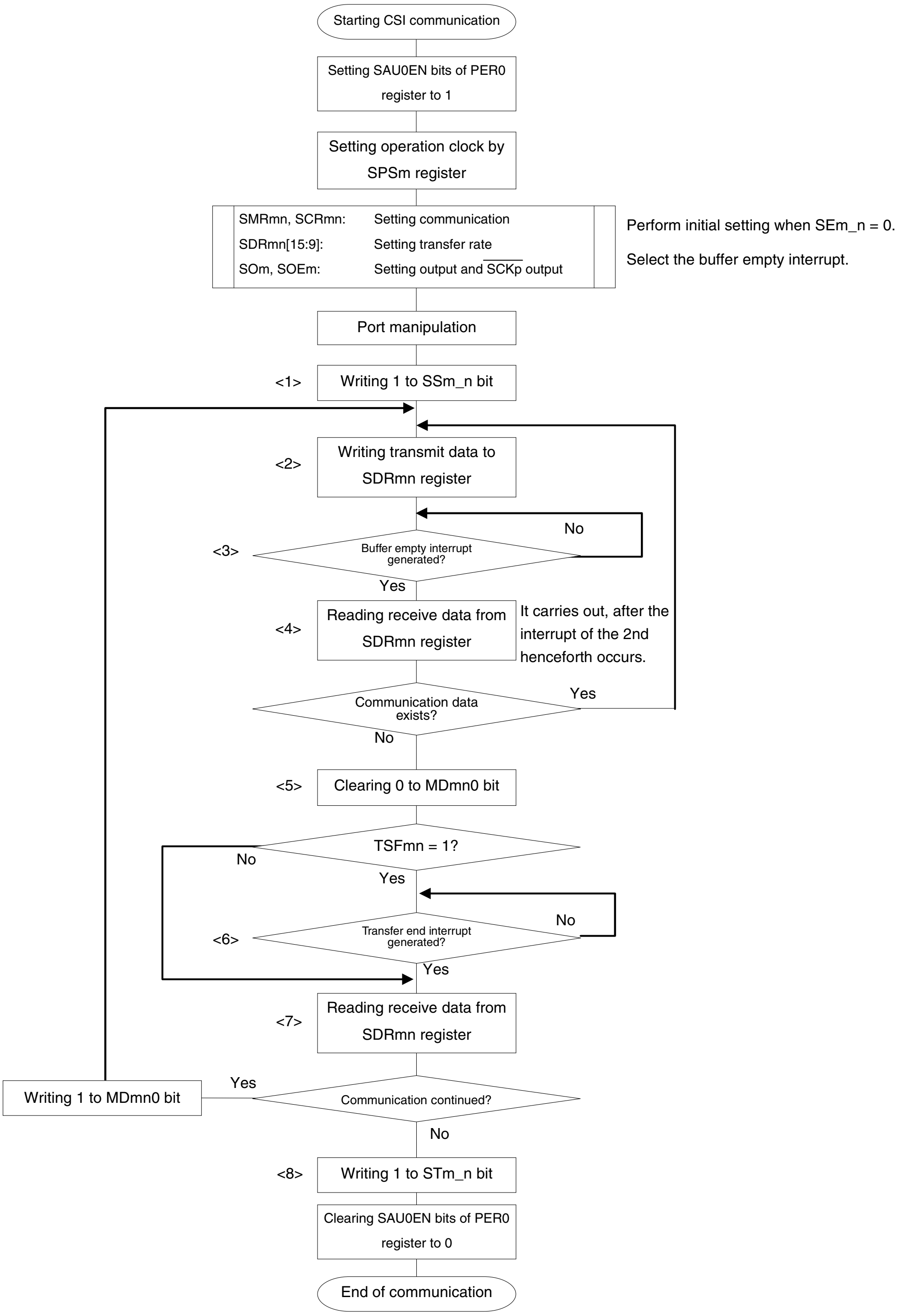
2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 11-99. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)

<R>

<R>

<R>


Starting CSI communication
Setting SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting transfer rate
SOm, SOEm: Setting output and $\overline{\text{SCKp}}$ output
Perform initial setting when SEm_n = 0.
Select the buffer empty interrupt.
Port manipulation
<1> Writing 1 to SSm_n bit
<2> Writing transmit data to SDRmn register
<3> Buffer empty interrupt generated?
No
Yes
<4> Reading receive data from SDRmn register
It carries out, after the interrupt of the 2nd henceforth occurs.
Communication data exists?
Yes
No
<5> Clearing 0 to MDmn0 bit
TSFmn = 1?
No
Yes
<6> Transfer end interrupt generated?
No
Yes
<7> Reading receive data from SDRmn register
Communication continued?
Yes
Writing 1 to MDmn0 bit
No
<8> Writing 1 to STm_n bit
Clearing SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-98 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

11.6.4 Slave transmission

Slave transmission is that the 78K0R/Hx3 transmits data to another device in the state of a transfer clock being input from another device.

SPI Function	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	$\overline{\text{SCK00}}$, SO00, $\overline{\text{SSI00}}$	$\overline{\text{SCK01}}$, SO01, $\overline{\text{SSI01}}$
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 to 16 bits	
Transfer rate	Max. $f_{MCK}/6$ [Hz][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	
SPI function	Slave select function operation selectable	

Notes 1. Because the external serial clock input to pins $\overline{\text{SCK00}}$ and $\overline{\text{SCK01}}$ is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-100. Example of Contents of Registers for Slave Transmission of SPI Function (CSI00, CSI01) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

(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	0	SOEm_1 0/1	SOEm_0 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	0	0	SSm_1 0/1	SSm_0 0/1

(d) 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	SCCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)

<R>

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

(f) Serial data register mn (SDRmn)

(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)
□: 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

Figure 11-100. Example of Contents of Registers for Slave Transmission of SPI Function (CSI00, CSI01) (2/2)

(ii) During operation (SEm_n = 1) (lower 8 bits: SDRpL)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Transmit data setting															
									SDRpL							

(g) Serial slave select enable register 0 (SSE0) ··· Control of the $\overline{\text{SSI}}$ pin of each slave channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSE0															SSE0_1	SSE0_0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

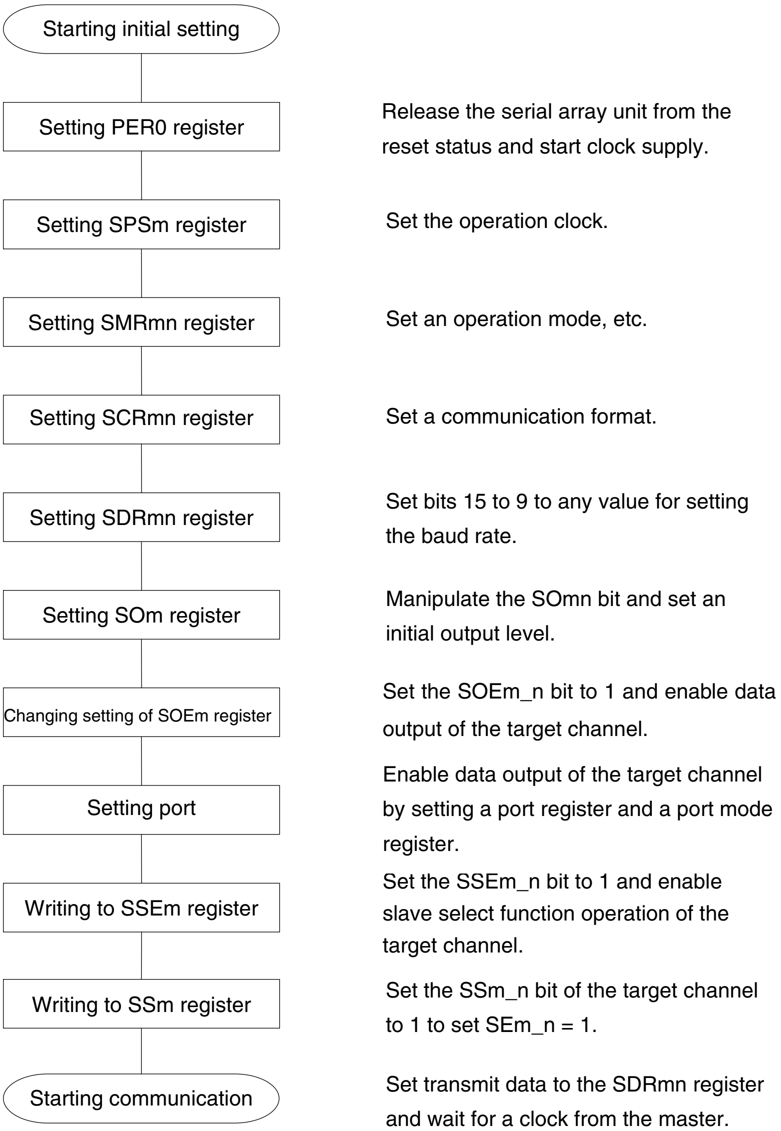
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

□ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-101. Initial Setting Procedure for Slave Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

<R> Set bits 15 to 9 to any value for setting the baud rate.

Manipulate the SOmn bit and set an initial output level.

Set the SOEm_n bit to 1 and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

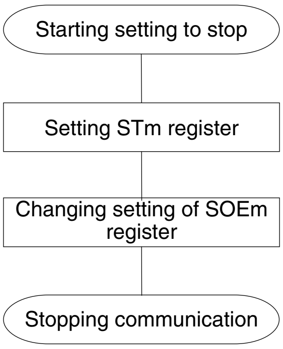
Set the SSEm_n bit to 1 and enable slave select function operation of the target channel.

Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Set transmit data to the SDRmn register and wait for a clock from the master.

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-102. Procedure for Stopping Slave Transmission



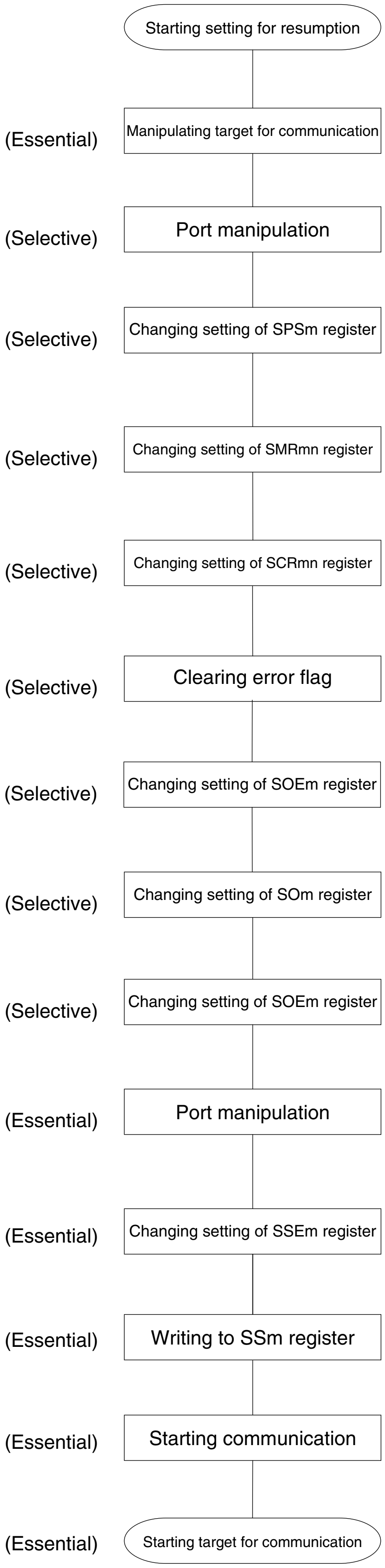
Write 1 to the STm_n bit of the target channel.

Clear the SOEm_n bit to 0 and stop the output of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-103 Procedure for Resuming Slave Transmission**).

Figure 11-103. Procedure for Resuming Slave Transmission


Starting setting for resumption
(Essential)
Manipulating target for communication
(Selective)
Port manipulation
(Selective)
Changing setting of SPSm register
(Selective)
Changing setting of SMRmn register
(Selective)
Changing setting of SCRmn register
(Selective)
Clearing error flag
(Selective)
Changing setting of SOEm register
(Selective)
Changing setting of SOm register
(Selective)
Changing setting of SOEm register
(Essential)
Port manipulation
(Essential)
Changing setting of SSEm register
(Essential)
Writing to SSm register
(Essential)
Starting communication
(Essential)
Starting target for communication

Stop the target for communication or wait until the target completes its operation.

Disable data output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if the setting of the SMRmn register is incorrect.

Change the setting if the setting of the SCRmn register is incorrect.

Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.

Set the SOEm register and stop the output of the target channel.

Manipulate the SOmn bit and set an initial output level.

Set the SOEm register and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

<R> Set the SSEm_n bit to 1 and enable slave select function operation of the target channel.

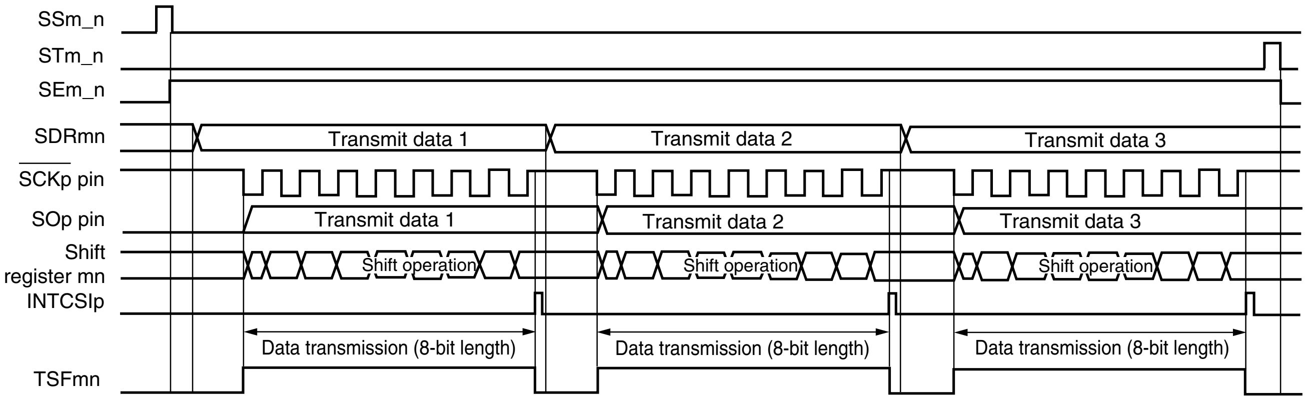
<R> Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Sets transmit data to the SDRmn register and wait for a clock from the master.

Starts the target for communication.

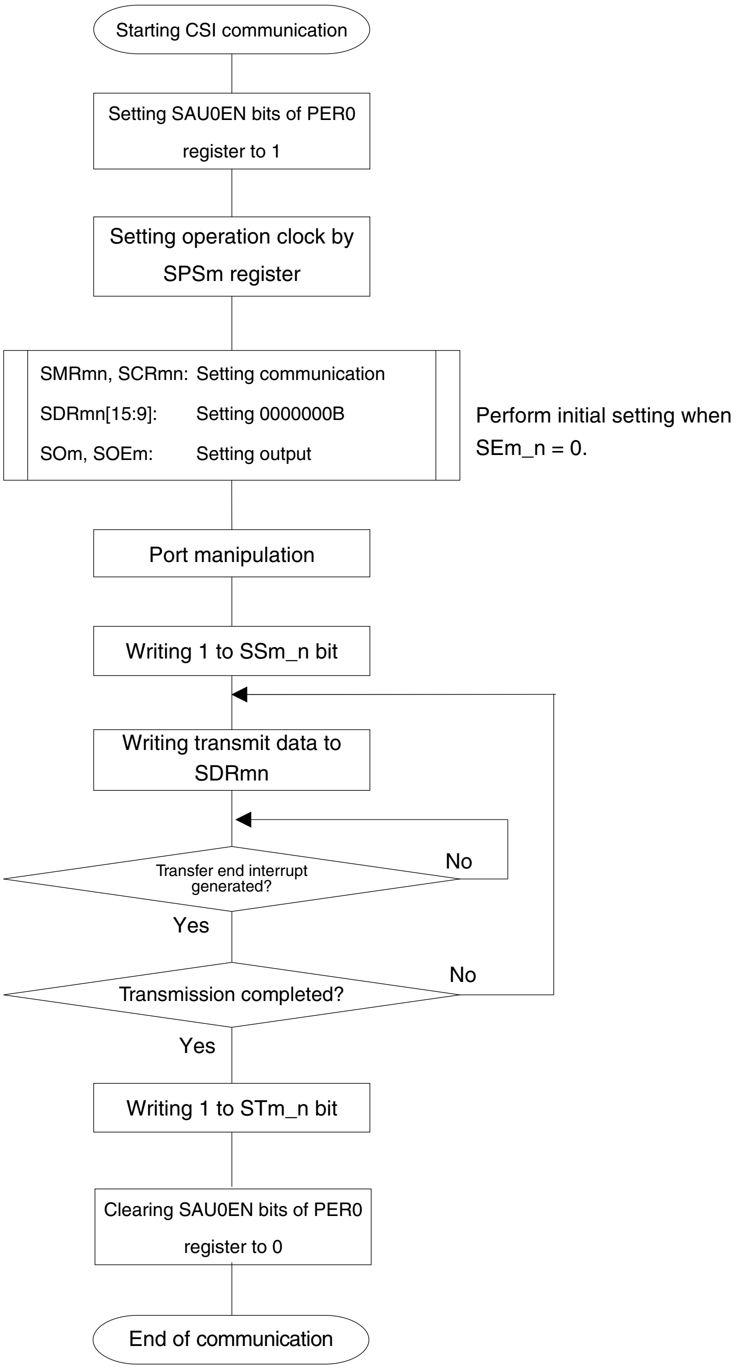
(3) Processing flow (in single-transmission mode)

Figure 11-104. Timing Chart of Slave Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

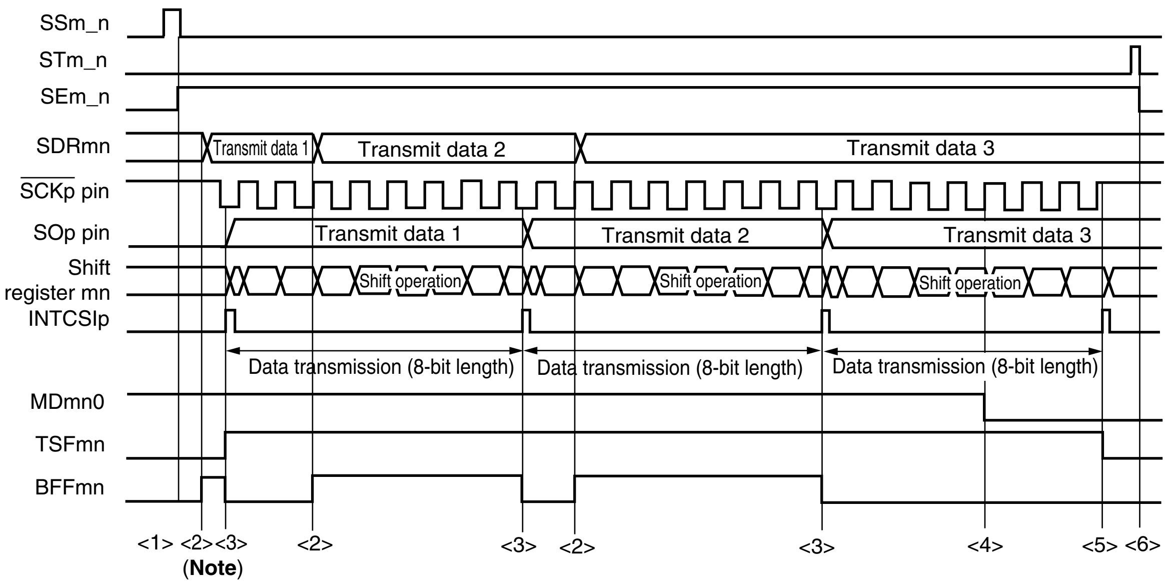
Figure 11-105. Flowchart of Slave Transmission (in Single-Transmission Mode)


Starting CSI communication
Setting SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting 0000000B
SOm, SOEm: Setting output
Perform initial setting when SEm_n = 0.
Port manipulation
Writing 1 to SSm_n bit
Writing transmit data to SDRmn
Transfer end interrupt generated?
No
Yes
Transmission completed?
No
Yes
Writing 1 to STm_n bit
Clearing SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

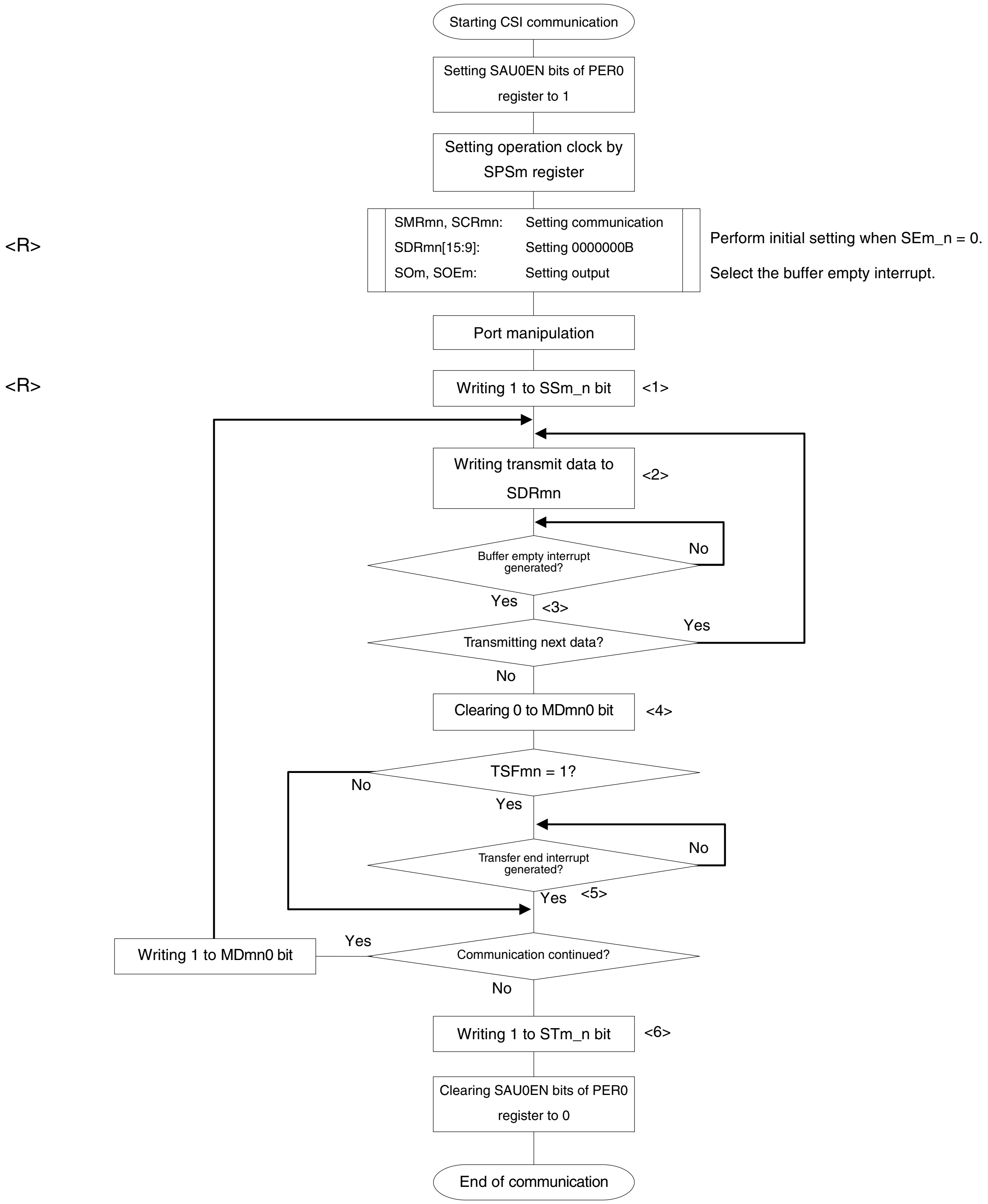
Figure 11-106. Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



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.

Figure 11-107. Flowchart of Slave Transmission (in Continuous Transmission Mode)


Starting CSI communication
Setting SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting 0000000B
SOm, SOEm: Setting output
Perform initial setting when SEm_n = 0.
Select the buffer empty interrupt.
<R>
Port manipulation
<R>
Writing 1 to SSm_n bit
<1>
Writing transmit data to SDRmn
<2>
Buffer empty interrupt generated?
No
Yes
<3>
Transmitting next data?
Yes
No
Clearing 0 to MDmn0 bit
<4>
TSFmn = 1?
No
Yes
Transfer end interrupt generated?
No
Yes
<5>
Writing 1 to MDmn0 bit
Yes
Communication continued?
No
Writing 1 to STm_n bit
<6>
Clearing SAU0EN bits of PER0 register to 0
End of communication

Caution **After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.**

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-106 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

11.6.5 Slave reception

Slave reception is that the 78K0R/Hx3 receives data from another device in the state of a transfer clock being input from another device.

SPI Function	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	$\overline{\text{SCK00}}$, SI00, $\overline{\text{SSI00}}$	$\overline{\text{SCK01}}$, SI01, $\overline{\text{SSI01}}$
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 to 16 bits	
Transfer rate	Max. f_{MCK}/6 [Hz]^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	
SPI function	Slave select function operation selectable	

Notes 1. Because the external serial clock input to pins $\overline{\text{SCK00}}$ and $\overline{\text{SCK01}}$ is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-108. Example of Contents of Registers for Slave Reception of SPI Function (CSI00, CSI01) (1/2)

(a) Serial output register m (SOm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 ×

<R> **(b) Serial output enable register m (SOEm) …Set the receive target channel to 0.**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm_1 0/1	SOEm_0 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	0	0	SSm_1 0/1	SSm_0 0/1

(d) 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	SCCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

Interrupt sources of channel n
0: Transfer end interrupt

(e) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<R> SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

(f) Serial data register mn (SDRmn)

(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)
□ : 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

Figure 11-108. Example of Contents of Registers for Slave Reception of SPI Function (CSI00, CSI01) (2/2)

(ii) During operation (SEm_n = 1) (lower 8 bits: SDRpL)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Receive data register															
									SDRpL							

(g) Serial slave select enable register 0 (SSE0) ··· Control of the $\overline{\text{SSI}}$ pin of each slave channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															SSE0_1	SSE0_0
SSE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

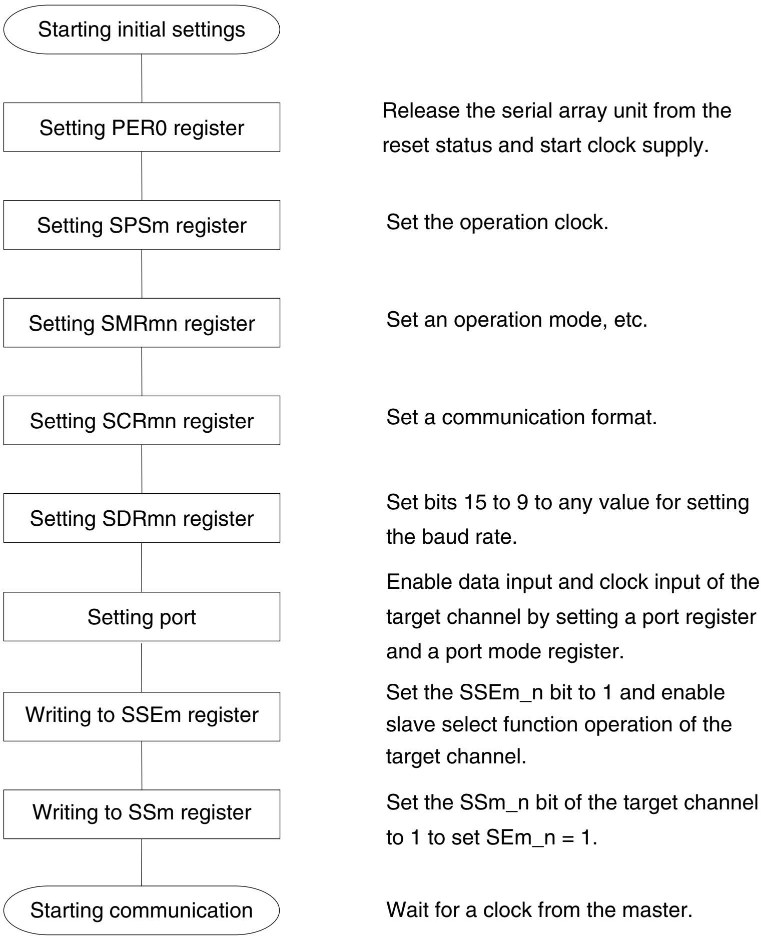
□ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-109. Initial Setting Procedure for Slave Reception

<R>



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to any value for setting the baud rate.

Enable data input and clock input of the target channel by setting a port register and a port mode register.

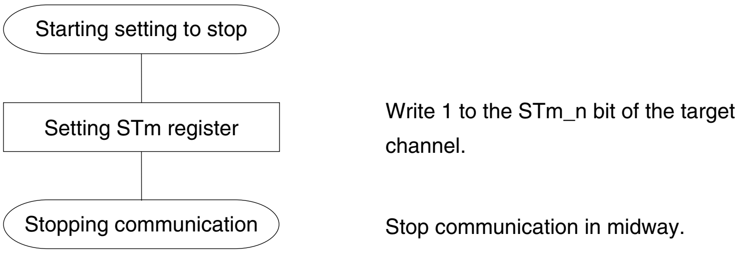
Set the SSEm_n bit to 1 and enable slave select function operation of the target channel.

Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Wait for a clock from the master.

Cautions After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

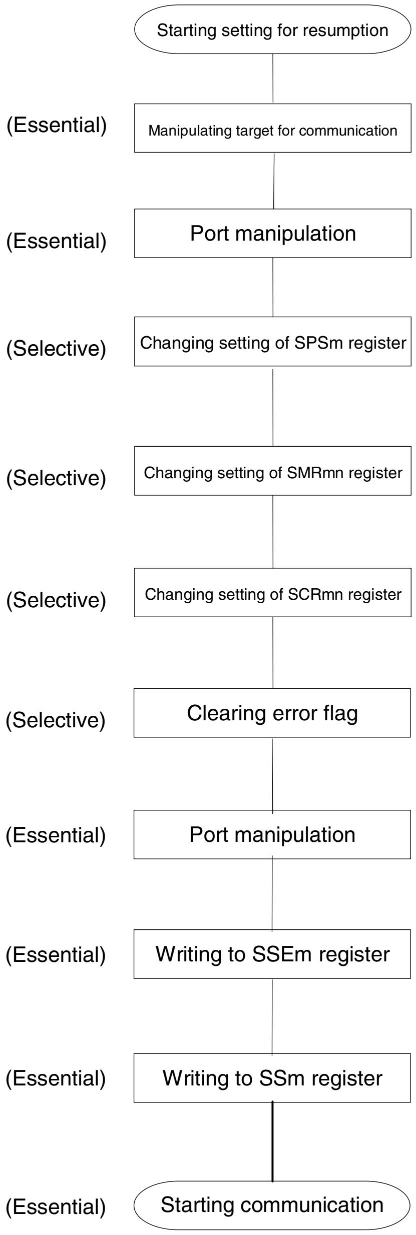
Figure 11-110. Procedure for Stopping Slave Reception



Write 1 to the STm_n bit of the target channel.

Stop communication in midway.

Figure 11-111. Procedure for Resuming Slave Reception



Stop the target for communication or wait until the target completes its operation.

Disable clock output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if the setting of the SMRmn register is incorrect.

Change the setting if the setting of the SCRmn register is incorrect.

Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.

Enable clock output of the target channel by setting a port register and a port mode register.

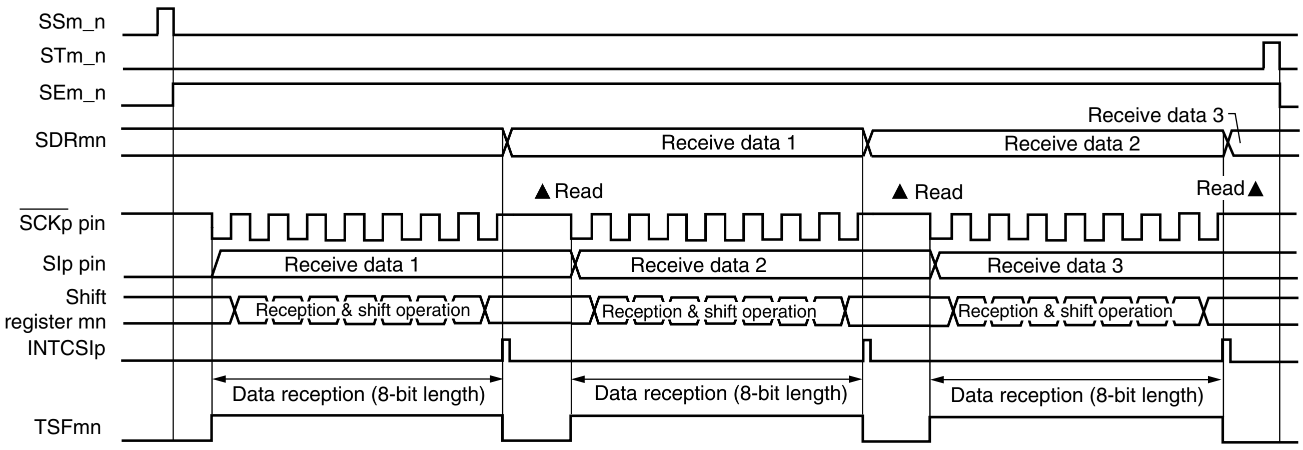
Set the SSEm_n bit to 1 and enable slave select function operation of the target channel.

Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Wait for a clock from the master.

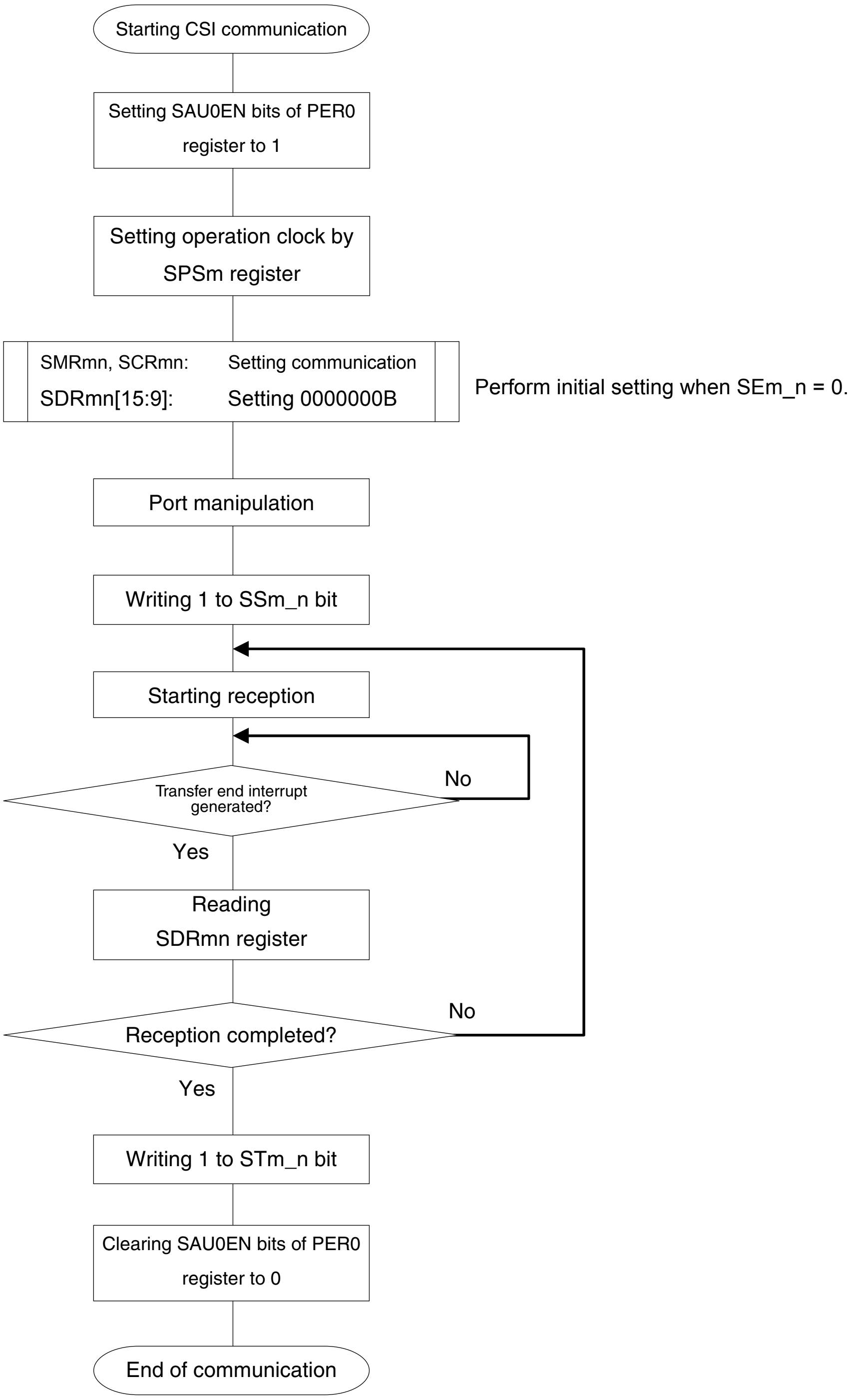
(3) Processing flow (in single-reception mode)

Figure 11-112. Timing Chart of Slave Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)


SSm_n
STm_n
SEm_n
SDRmn
Receive data 1
Receive data 2
Receive data 3
▲Read
▲Read
Read▲
$\overline{\text{SCKp}}$ pin
SIp pin
Receive data 1
Receive data 2
Receive data 3
Shift register mn
Reception & shift operation
Reception & shift operation
Reception & shift operation
INTCSIp
Data reception (8-bit length)
Data reception (8-bit length)
Data reception (8-bit length)
TSFmn

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 11-113. Flowchart of Slave Reception (in Single-Reception Mode)


Starting CSI communication
Setting SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting 0000000B
Perform initial setting when SEm_n = 0.
Port manipulation
Writing 1 to SSm_n bit
Starting reception
Transfer end interrupt generated?
No
Yes
Reading SDRmn register
Reception completed?
No
Yes
Writing 1 to STm_n bit
Clearing SAU0EN bits of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

11.6.6 Slave transmission/reception

Slave transmission/reception is that the 78K0R/Hx3 transmits/receives data to/from another device in the state of a transfer clock being input from another device.

SPI Function	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	$\overline{\text{SCK00}}$, SI00, SO00, $\overline{\text{SSI00}}$	$\overline{\text{SCK01}}$, SI01, SO01, $\overline{\text{SSI01}}$
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 to 16 bits	
Transfer rate	Max. $f_{\text{MCK}}/6$ [Hz][Notes 1, 2]	
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data input/output starts at the start of the operation of the serial clock. • DAPmn = 1: Data input/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	
SPI function	Slave select function operation selectable	

Notes 1. Because the external serial clock input to pins $\overline{\text{SCK00}}$ and $\overline{\text{SCK01}}$ is sampled internally and used, the fastest transfer rate is $f_{\text{MCK}}/6$ [Hz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-114. Example of Contents of Registers for Slave Transmission/Reception of SPI Function (CSI00, CSI01) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

(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	0	SOEm_1 0/1	SOEm_0 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	0	0	SSm_1 0/1	SSm_0 0/1

(d) 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	SCCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)

		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<R>	SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

(f) Serial data register mn (SDRmn)

(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)
□ : 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

Figure 11-114. Example of Contents of Registers for Slave Transmission/Reception of SPI Function (CSI00, CSI01) (2/2)

(ii) During operation (SEm_n = 1) (lower 8 bits: SDRpL)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Transmit data setting/receive data register															
									SDRpL							

(g) Serial slave select enable register m (SSEm) ··· Control of the $\overline{\text{SSI}}$ pin of each slave channel.

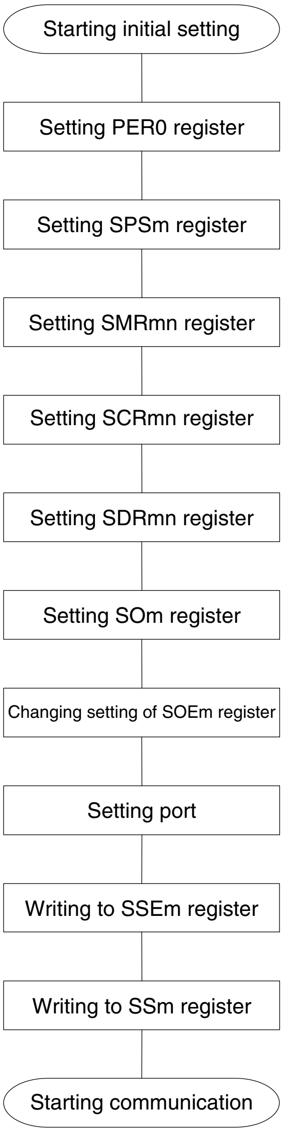
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSEm															SSEm_1	SSEm_0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)
□ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-115. Initial Setting Procedure for Slave Transmission/Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

<R> Set bits 15 to 9 to any value for setting the baud rate.

Manipulate the SOmn bit and set an initial output level.

Set the SOEm_n bit to 1 and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

Set the SSEm_n bit to 1 and enable slave select function operation of the target channel.

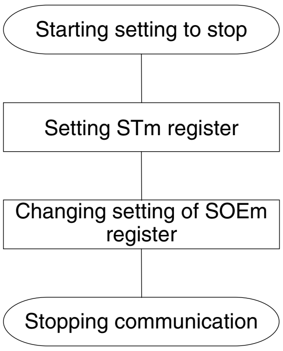
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

Set transmit data to the SDRmn register and wait for a clock from the master.

Cautions 1. After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

2. Be sure to set transmit data to the SDRpL register before the clock from the master is started.

Figure 11-116. Procedure for Stopping Slave Transmission/Reception



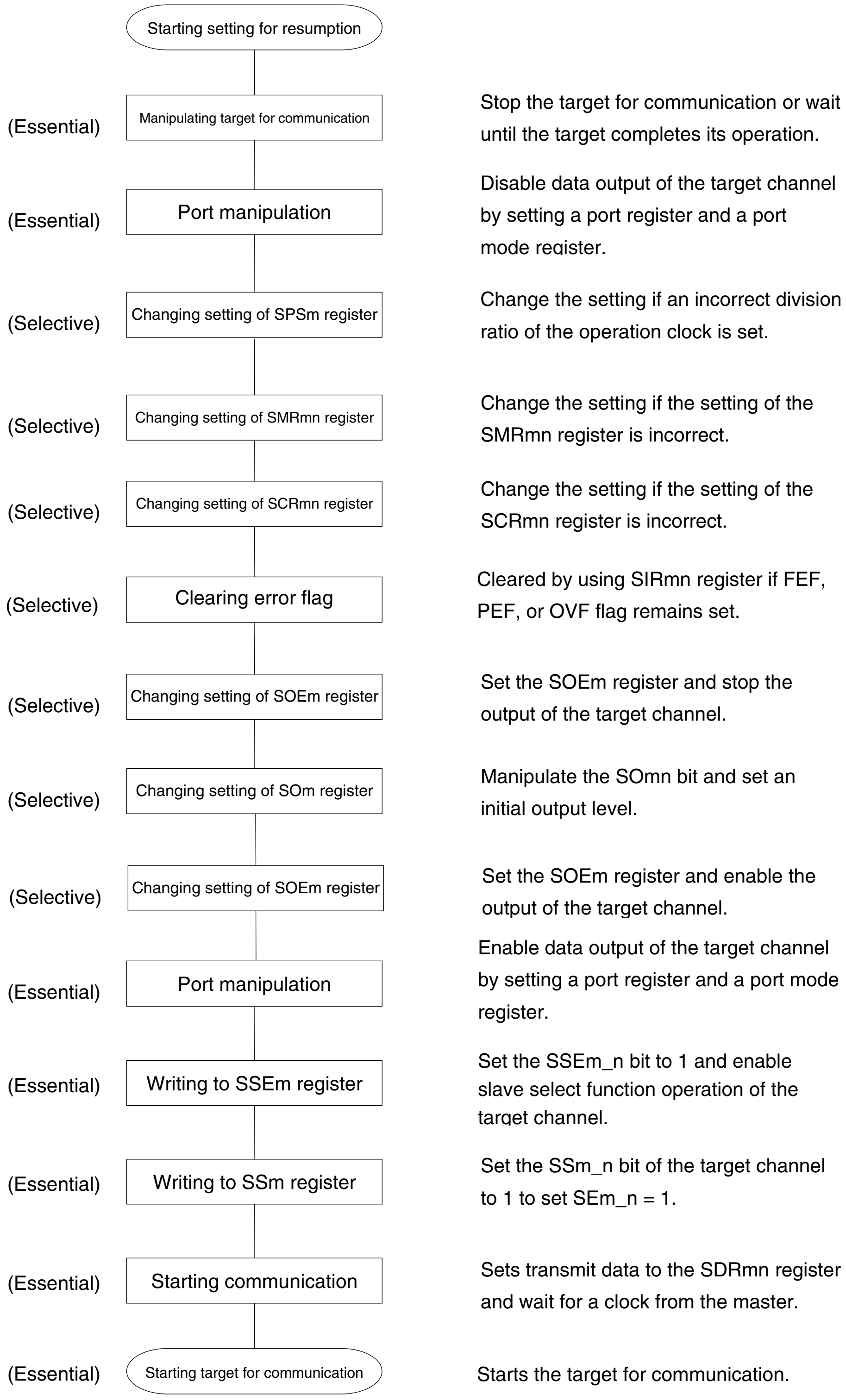
Write 1 to the STm_n bit of the target channel.

Clear the SOEm_n bit to 0 and stop the output of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-117 Procedure for Resuming Slave Transmission/Reception**).

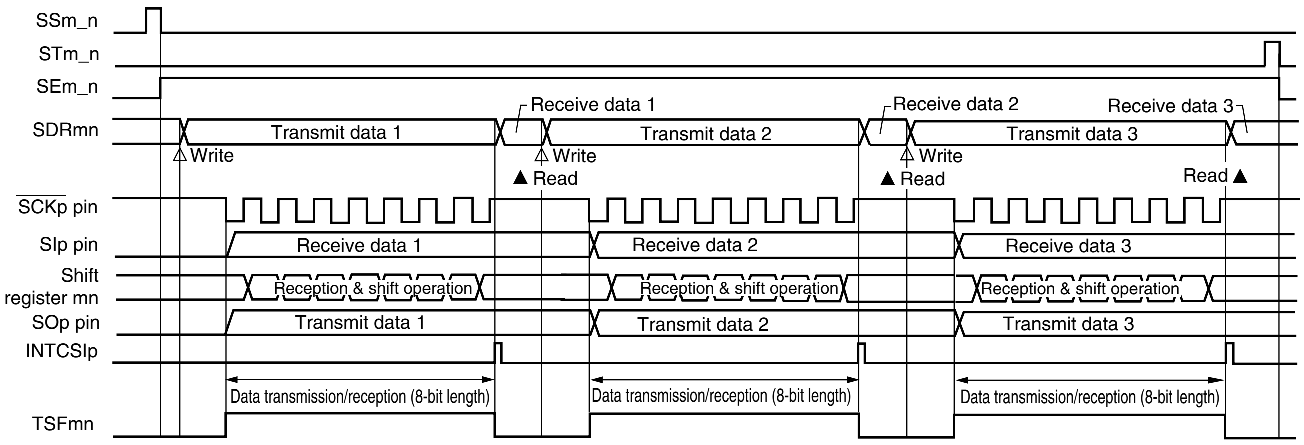
Figure 11-117. Procedure for Resuming Slave Transmission/Reception


Starting setting for resumption
(Essential)
Manipulating target for communication
Stop the target for communication or wait until the target completes its operation.
(Essential)
Port manipulation
Disable data output of the target channel by setting a port register and a port mode register.
(Selective)
Changing setting of SPSm register
Change the setting if an incorrect division ratio of the operation clock is set.
(Selective)
Changing setting of SMRmn register
Change the setting if the setting of the SMRmn register is incorrect.
(Selective)
Changing setting of SCRmn register
Change the setting if the setting of the SCRmn register is incorrect.
(Selective)
Clearing error flag
Cleared by using SIRmn register if FEF, PEF, or OVF flag remains set.
(Selective)
Changing setting of SOEm register
Set the SOEm register and stop the output of the target channel.
(Selective)
Changing setting of SOm register
Manipulate the SOmn bit and set an initial output level.
(Selective)
Changing setting of SOEm register
Set the SOEm register and enable the output of the target channel.
(Essential)
Port manipulation
Enable data output of the target channel by setting a port register and a port mode register.
(Essential)
Writing to SSEm register
Set the SSEm_n bit to 1 and enable slave select function operation of the target channel.
(Essential)
Writing to SSm register
Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.
(Essential)
Starting communication
Sets transmit data to the SDRmn register and wait for a clock from the master.
(Essential)
Starting target for communication
Starts the target for communication.

Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.

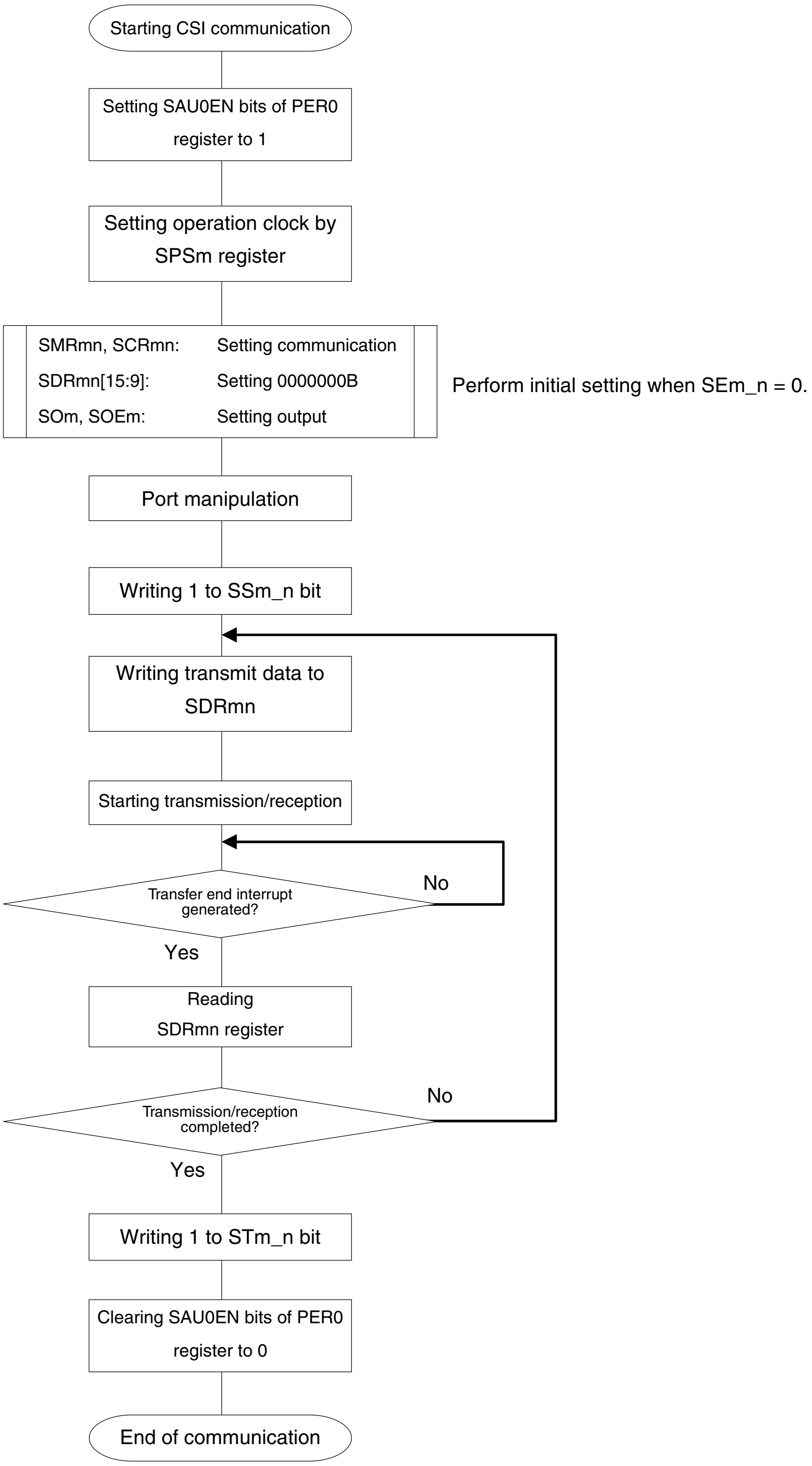
(3) Processing flow (in single-transmission/reception mode)

Figure 11-118. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 11-119. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)

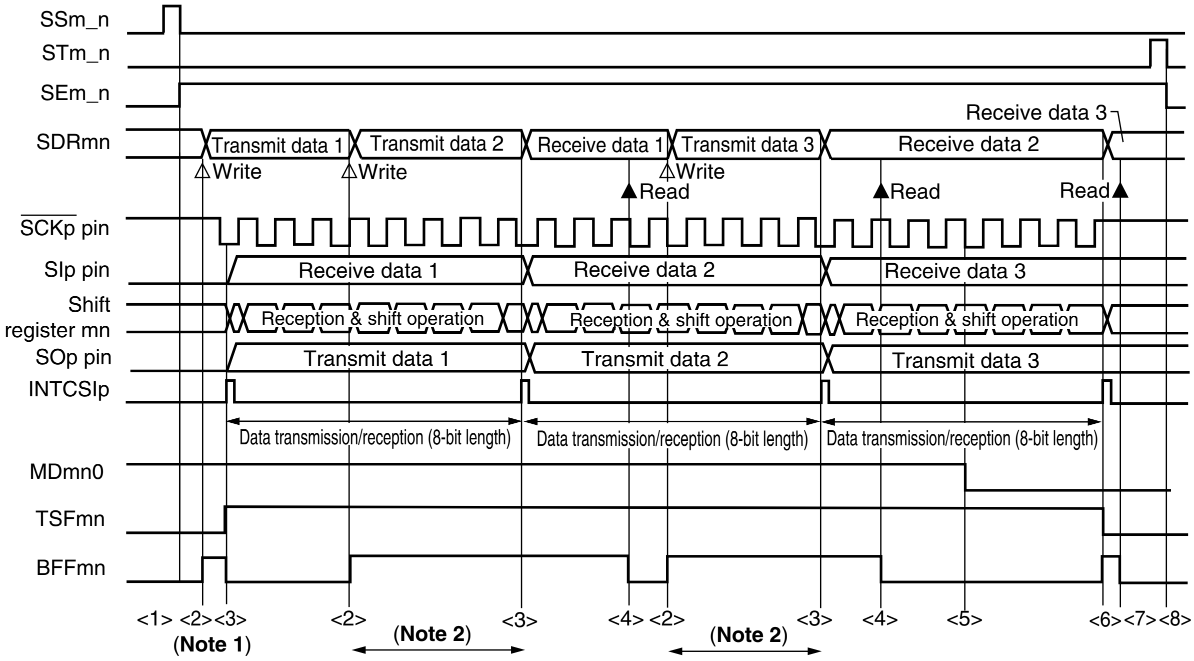

Starting CSI communication
Setting SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting 0000000B
SOm, SOEm: Setting output
Perform initial setting when SEm_n = 0.
Port manipulation
Writing 1 to SSm_n bit
Writing transmit data to SDRmn
Starting transmission/reception
Transfer end interrupt generated?
No
Yes
Reading SDRmn register
Transmission/reception completed?
No
Yes
Writing 1 to STm_n bit
Clearing SAU0EN bits of PER0 register to 0
End of communication

Cautions 1. After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

2. Be sure to set transmit data to the SDRpL register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-120. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



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 11-121 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

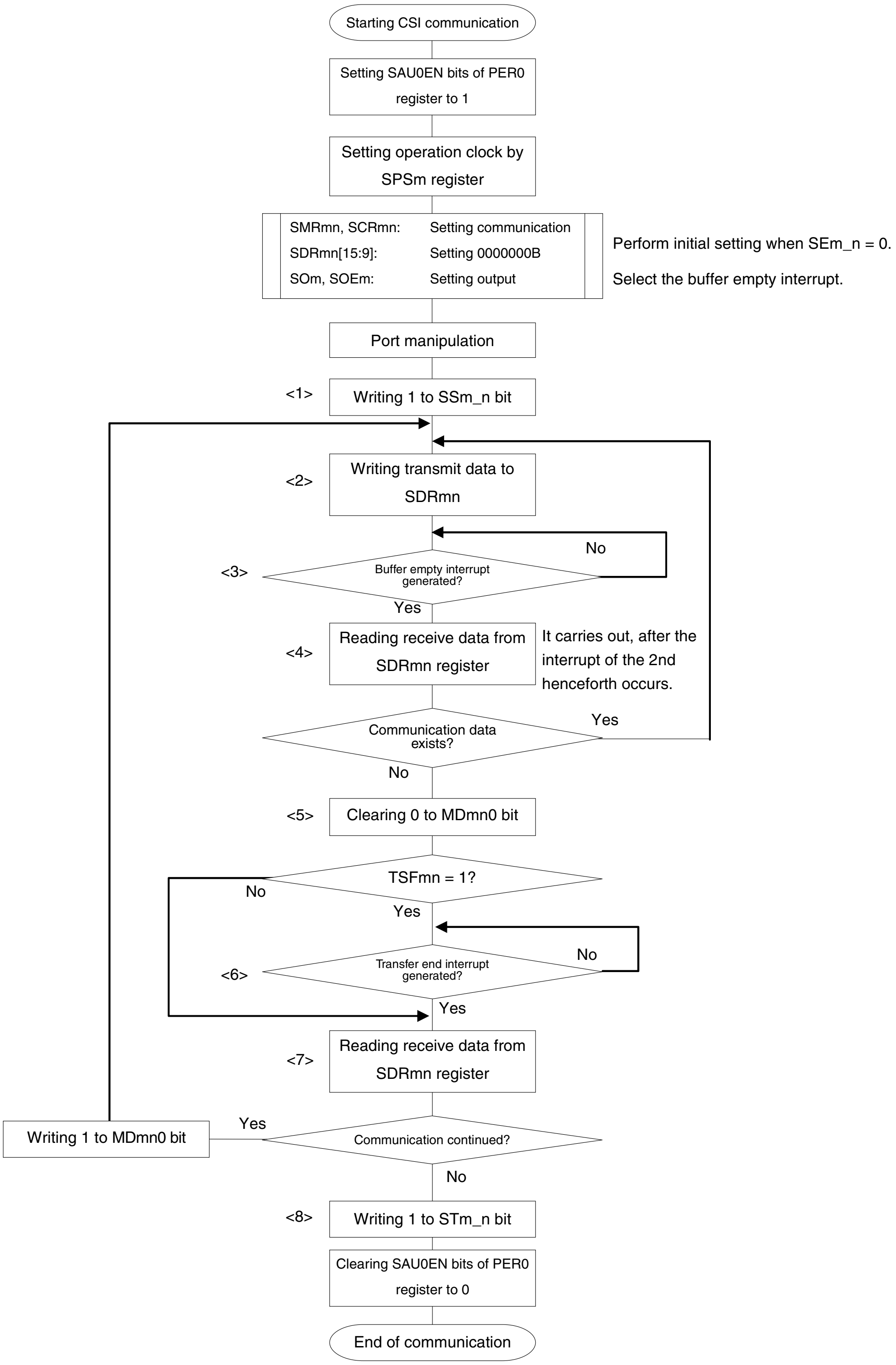
Figure 11-121. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)

<R>

<R>

<R>

<R>


Starting CSI communication
Setting SAU0EN bits of PER0 register to 1
Setting operation clock by SPSm register
SMRmn, SCRmn: Setting communication
SDRmn[15:9]: Setting 0000000B
SOm, SOEm: Setting output
Perform initial setting when SEm_n = 0.
Select the buffer empty interrupt.
Port manipulation
<1>
Writing 1 to SSm_n bit
<2>
Writing transmit data to SDRmn
<3>
Buffer empty interrupt generated?
No
Yes
<4>
Reading receive data from SDRmn register
It carries out, after the interrupt of the 2nd henceforth occurs.
Communication data exists?
Yes
No
<5>
Clearing 0 to MDmn0 bit
TSFmn = 1?
No
Yes
<6>
Transfer end interrupt generated?
No
Yes
<7>
Reading receive data from SDRmn register
Communication continued?
Yes
Writing 1 to MDmn0 bit
No
<8>
Writing 1 to STm_n bit
Clearing SAU0EN bits of PER0 register to 0
End of communication

Cautions 1. After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

2. Be sure to set transmit data to the SDRpL register before the clock from the master is started.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-120 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

11.6.7 Calculating transfer clock frequency

The transfer clock frequency for SPI function (CSI00, CSI01) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) [Hz] = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

(2) Slave

(Transfer clock frequency) [Hz] = {Frequency of serial clock (f_{SCK}) supplied by master}[Note]

Note The permissible maximum transfer clock frequency is $f_{MCK}/6$.

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 = 0), n: Channel number (n = 0, 1)

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11-5. Operating Clock Selection

SMRmn Register	SPSm Register								Operation Clock (f_{MCK})[Note]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	24 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	12 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	93.75 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	46.86 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	23.44 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	11.72 kHz
	X	X	X	X	1	1	1	1	INTTM23	
1	0	0	0	0	X	X	X	X	f_{CLK}	24 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	12 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	93.75 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	46.86 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	23.44 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	11.72 kHz
	1	1	1	1	X	X	X	X	INTTM23	
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (STm = 0003H) the operation of the serial array unit m (SAUm). When selecting INTTM23 for the operation clock, also stop the timer array unit 2 (TAU2) (TT2 = 00FFH).

Remarks 1. X: Don’t care

2. m: Unit number (m = 0), n: Channel number (n = 0, 1)

11.6.8 Procedure for processing errors that occurred during SPI Function (CSI00, CSI01) communication

The procedure for processing errors that occurred during SPI function (CSI00, CSI01) communication is described in Figure 11-122.

Figure 11-122. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data SDRmn register. →	The 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.	Only error generated at the point of reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

11.7 Operation of UART 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 7 to 9 bits or 16 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 detection flag]

- Framing error, parity error, or overrun error

UART2 uses channels 0 and 1 of SAU2.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^{2}C
0	0	CSI00 (supports SPI)	–	–
	1	CSI01 (supports SPI)		–
1	0	CSI20	–	–
	1	CSI21		IIC11
2	0	–	UART2	IIC20
	1	–		–

Caution When using serial array unit as UART, the channels of both the transmitting side (even-number channel) and the receiving side (odd-number channel) can be used only as UART.

UART performs the following two types of communication operations.

- UART transmission (See **11.7.1**.)
- UART reception (See **11.7.2**.)

11.7.1 UART transmission

UART transmission is an operation to transmit data from the 78K0R/Hx3 to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART2
Target channel	Channel 0 of SAU2
Pins used	TxD2
Interrupt	INTST2, 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 to 9 bits or 16 bits
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR2n [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 29 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-123. Example of Contents of Registers for UART Transmission of UART2 (1/2)

(a) Serial output register 2 (SO2) … Sets only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO2	0	0	0	0	0	0	CKO21 ×	CKO20 ×	0	0	0	0	0	0	SO21 ×	SO20 0/1 Note

(b) Serial output enable register 2 (SOE2)

		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<R>	SOE2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE2_1 ×	SOE2_0 1

(c) Serial channel start register 2 (SS2)

		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<R>	SS2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS2_1 ×	SS2_0 1

(d) Serial output level register 2 (SOL2) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL20 0/1

0: Forward (normal) transmission
1: Reverse transmission

(e) Serial mode register 20 (SMR20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR20	CKS20 0/1	SCCS20 0	0	0	0	0	0	STS20 0	0	SIS200 0	1	0	0	MD202 0	MD201 1	MD200 0/1

Interrupt sources of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

Note Before transmission is started, be sure to set to 1 when the SOL20 bit of the target channel is set to 0, and set to 0 when the SOL20 bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remark □ : 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 11-123. Example of Contents of Registers for UART Transmission of UART2 (2/2)

(f) Serial communication operation setting register 20 (SCR20)

<R>

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR20	TXE20 1	RXE20 0	DAP20 0	CKP20 0	0	0	PTC201 0/1	PTC200 0/1	DIR20 0/1	0	SLC201 0/1	SLC200 0/1	DLS203 0/1	DLS202 0/1	DLS201 0/1	DLS200 0/1

Setting of parity bit
00B: No parity
01B: 0 parity
10B: Even parity
11B: Odd parity

Setting of stop bit
01B: Appending 1 bit
10B: Appending 2 bits

(g) Serial data register mn (SDR20)

(i) When operation is stopped (SE2_0 = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR20	Baud rate setting							0	0	0	0	0	0	0	0	0

(ii) During operation (SE2_0 = 1) (lower 8 bits: SDR20L)

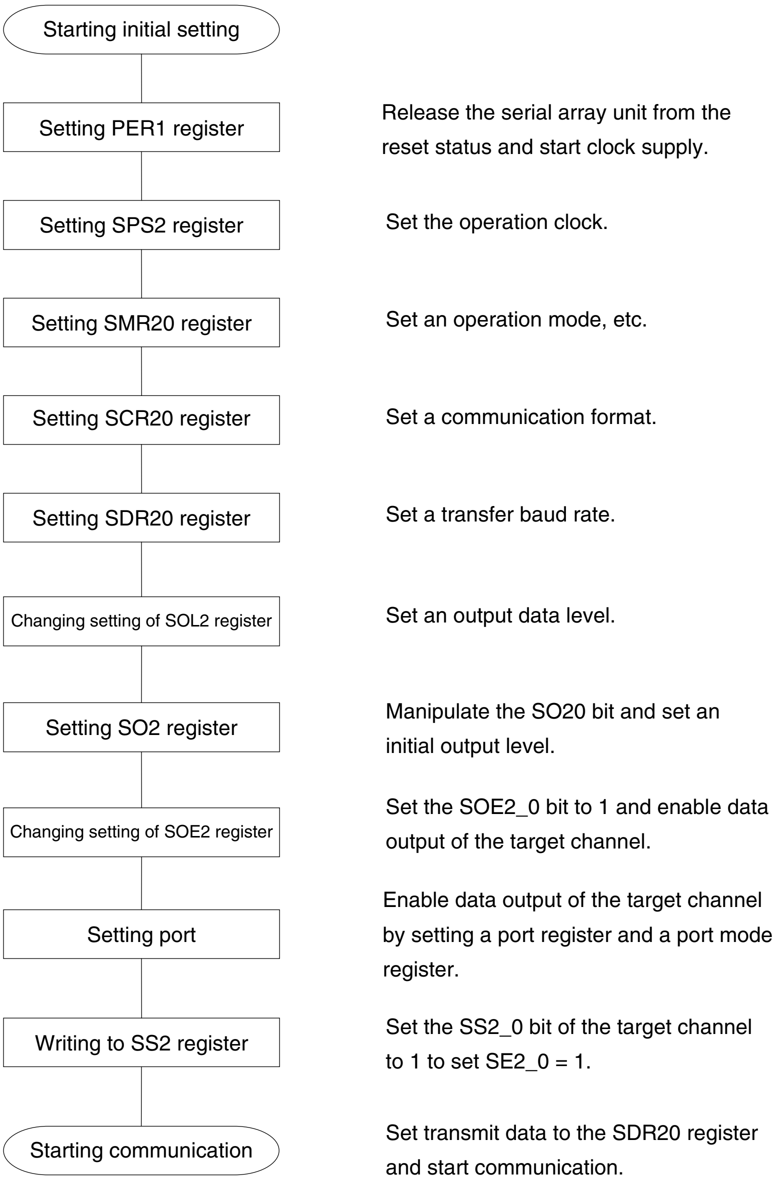
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR20	Transmit data setting															

SDR20L

Remark □: Setting is fixed in the UART transmission mode, ▨: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

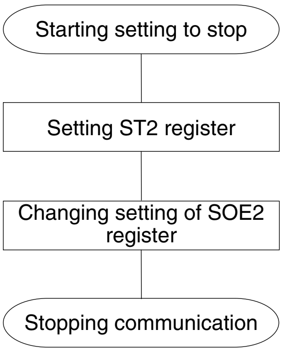
(2) Operation procedure

Figure 11-124. Initial Setting Procedure for UART Transmission


Starting initial setting
Setting PER1 register
Release the serial array unit from the reset status and start clock supply.
Setting SPS2 register
Set the operation clock.
Setting SMR20 register
Set an operation mode, etc.
Setting SCR20 register
Set a communication format.
Setting SDR20 register
Set a transfer baud rate.
Changing setting of SOL2 register
Set an output data level.
Setting SO2 register
Manipulate the SO20 bit and set an initial output level.
Changing setting of SOE2 register
Set the SOE2_0 bit to 1 and enable data output of the target channel.
Setting port
Enable data output of the target channel by setting a port register and a port mode register.
Writing to SS2 register
Set the SS2_0 bit of the target channel to 1 to set SE2_0 = 1.
Starting communication
Set transmit data to the SDR20 register and start communication.

Caution After setting the PER0 register to 1, be sure to set the SPS2 register after 4 or more clocks have elapsed.

Figure 11-125. Procedure for Stopping UART Transmission



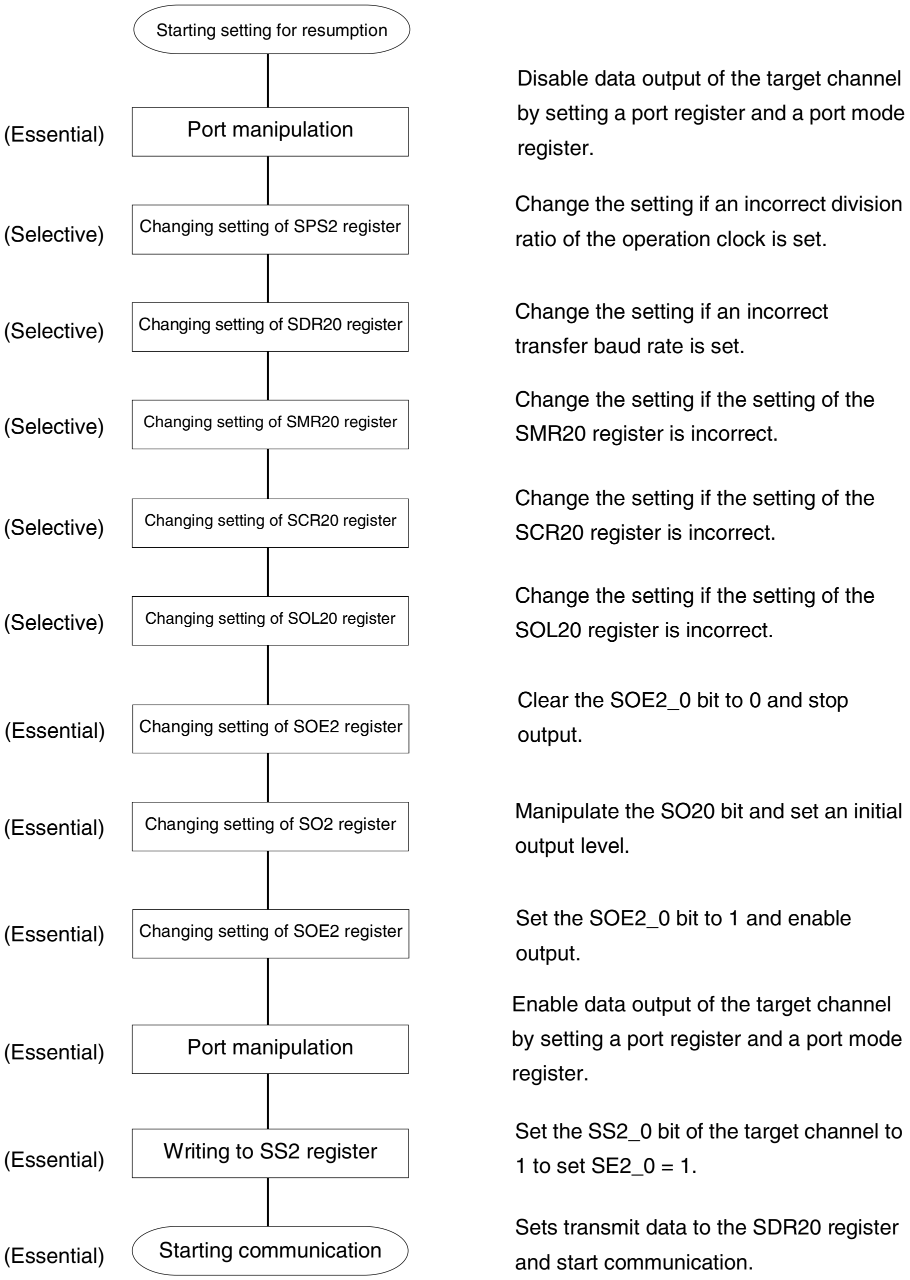
Write 1 to the ST2_0 bit of the target channel.

Clear the SOE2_0 bit to 0 and stop the output.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO2 register (see **Figure 11-126 Procedure for Resuming UART Transmission**).

Figure 11-126. Procedure for Resuming UART Transmission


Starting setting for resumption
(Essential)
Port manipulation
Disable data output of the target channel by setting a port register and a port mode register.
(Selective)
Changing setting of SPS2 register
Change the setting if an incorrect division ratio of the operation clock is set.
(Selective)
Changing setting of SDR20 register
Change the setting if an incorrect transfer baud rate is set.
(Selective)
Changing setting of SMR20 register
Change the setting if the setting of the SMR20 register is incorrect.
(Selective)
Changing setting of SCR20 register
Change the setting if the setting of the SCR20 register is incorrect.
(Selective)
Changing setting of SOL20 register
Change the setting if the setting of the SOL20 register is incorrect.
(Essential)
Changing setting of SOE2 register
Clear the SOE2_0 bit to 0 and stop output.
(Essential)
Changing setting of SO2 register
Manipulate the SO20 bit and set an initial output level.
(Essential)
Changing setting of SOE2 register
Set the SOE2_0 bit to 1 and enable output.
(Essential)
Port manipulation
Enable data output of the target channel by setting a port register and a port mode register.
(Essential)
Writing to SS2 register
Set the SS2_0 bit of the target channel to 1 to set SE2_0 = 1.
(Essential)
Starting communication
Sets transmit data to the SDR20 register and start communication.

(3) Processing flow (in single-transmission mode)

Figure 11-127. Timing Chart of UART Transmission (in Single-Transmission Mode)

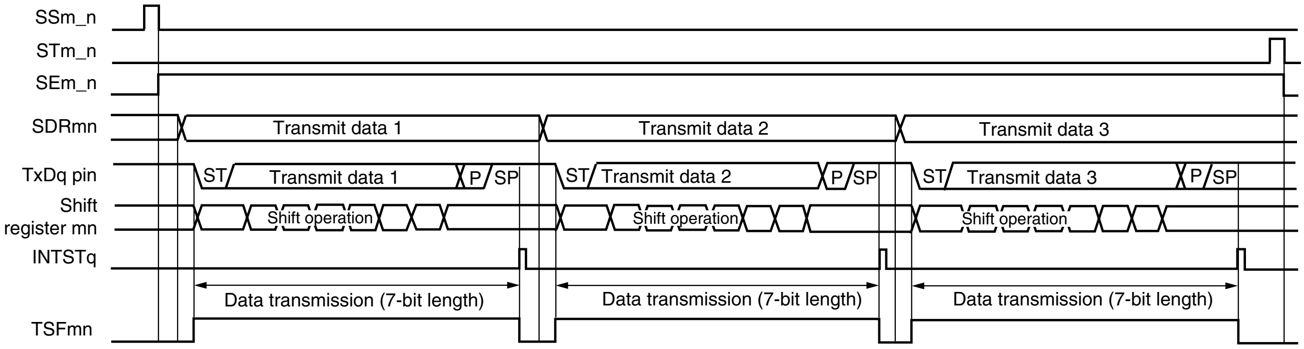
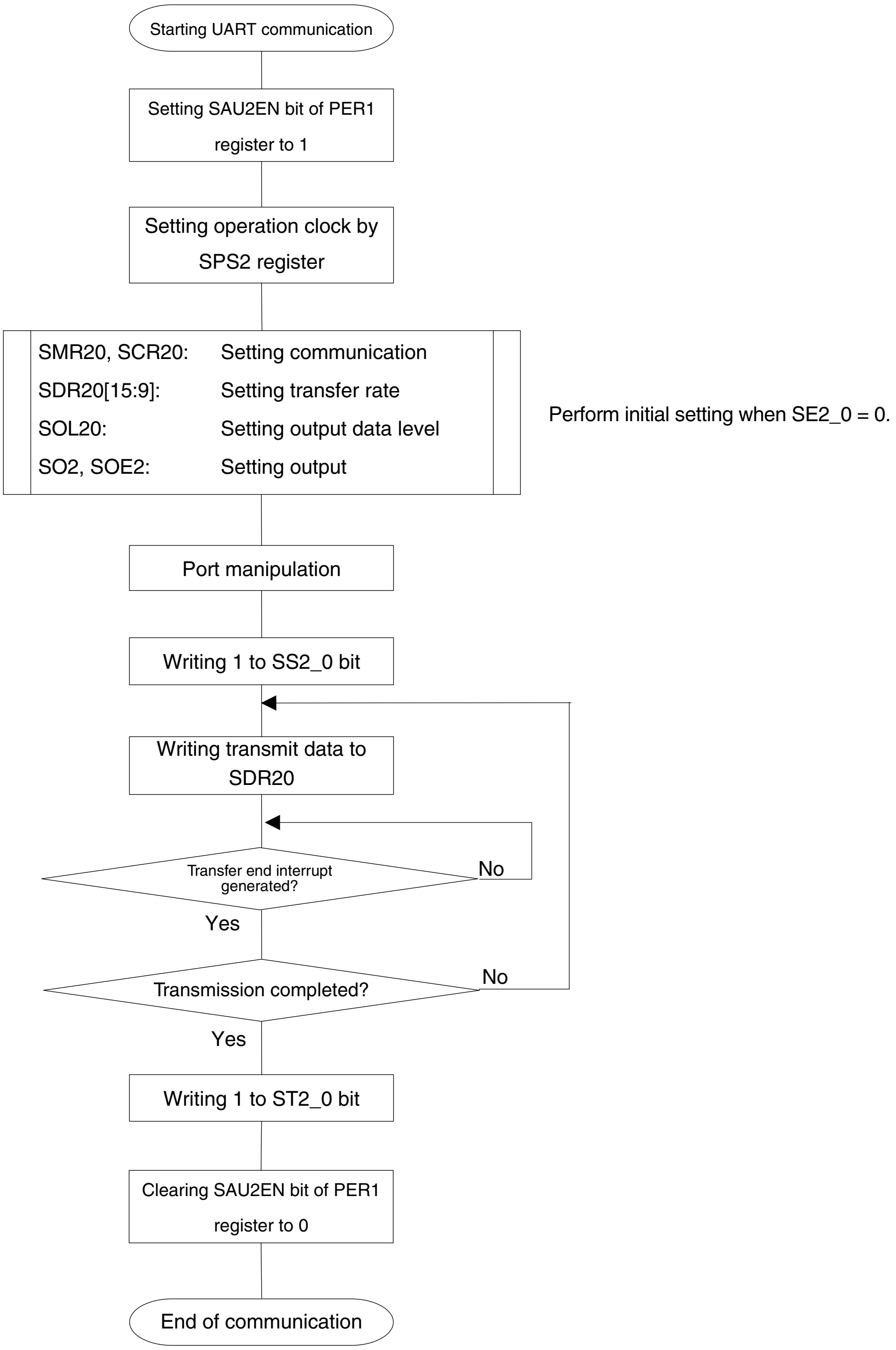


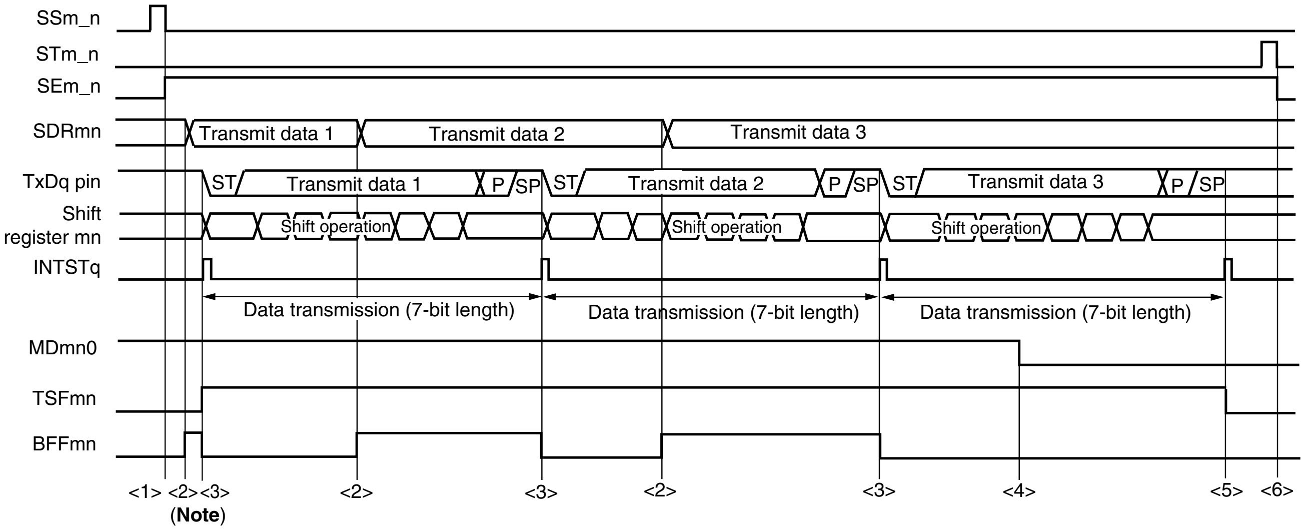
Figure 11-128. Flowchart of UART Transmission (in Single-Transmission Mode)


Starting UART communication
Setting SAU2EN bit of PER1 register to 1
Setting operation clock by SPS2 register
SMR20, SCR20: Setting communication
SDR20[15:9]: Setting transfer rate
SOL20: Setting output data level
SO2, SOE2: Setting output
Perform initial setting when SE2_0 = 0.
Port manipulation
Writing 1 to SS2_0 bit
Writing transmit data to SDR20
Transfer end interrupt generated?
No
Yes
Transmission completed?
No
Yes
Writing 1 to ST2_0 bit
Clearing SAU2EN bit of PER1 register to 0
End of communication

Caution After setting the PER1 register to 1, be sure to set the SPS2 register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 11-129. Timing Chart of UART Transmission (in Continuous Transmission Mode)



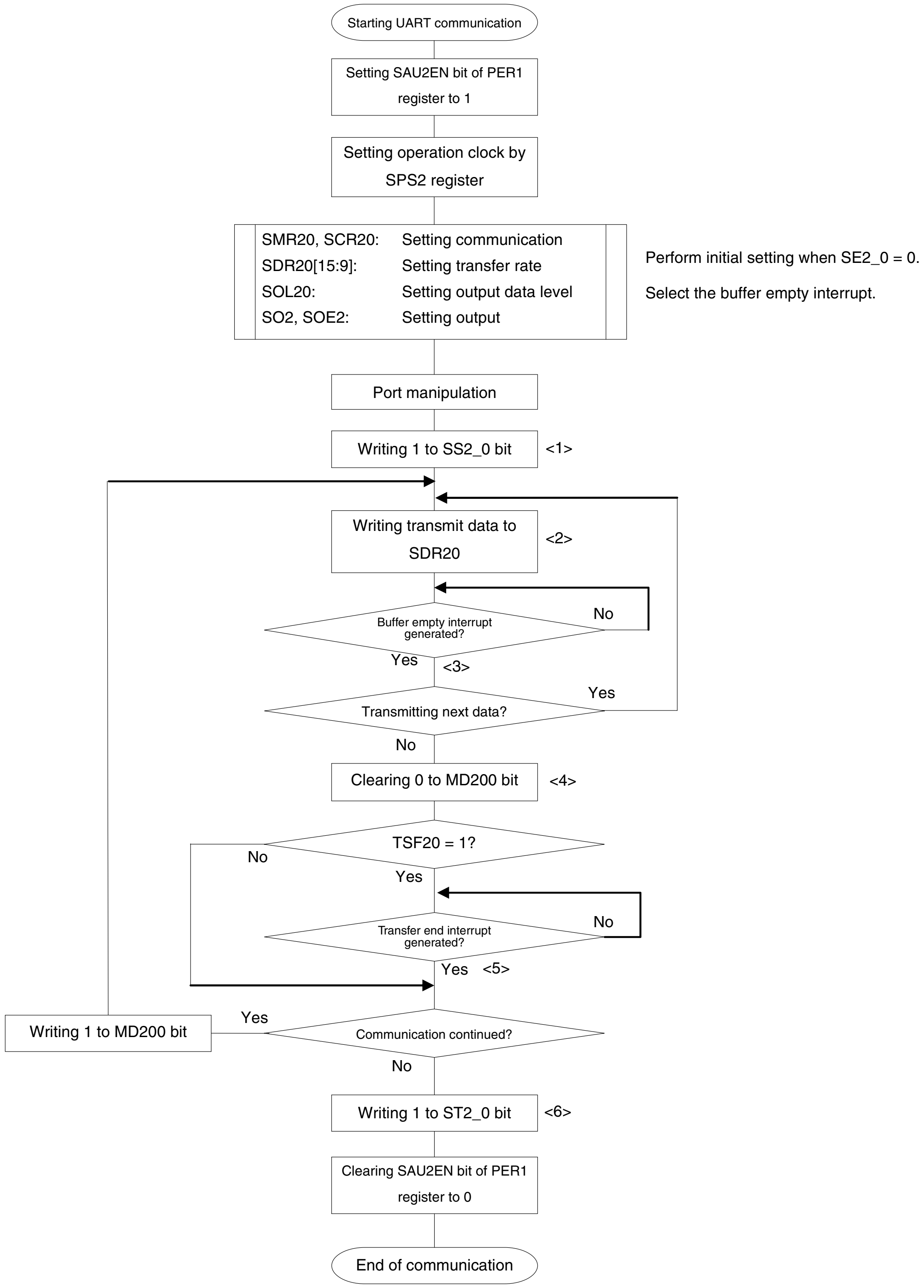
Note When transmit data is written to the SDR20register while BFF20 = 1, the transmit data is overwritten.

Caution The MD200 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.

Figure 11-130. Flowchart of UART Transmission (in Continuous Transmission Mode)

<R>

<R>


Starting UART communication
Setting SAU2EN bit of PER1 register to 1
Setting operation clock by SPS2 register
SMR20, SCR20: Setting communication
SDR20[15:9]: Setting transfer rate
SOL20: Setting output data level
SO2, SOE2: Setting output
Perform initial setting when SE2_0 = 0.
Select the buffer empty interrupt.
Port manipulation
Writing 1 to SS2_0 bit
<1>
Writing transmit data to SDR20
<2>
Buffer empty interrupt generated?
No
Yes
<3>
Transmitting next data?
Yes
No
Clearing 0 to MD200 bit
<4>
TSF20 = 1?
No
Yes
Transfer end interrupt generated?
No
Yes
<5>
Writing 1 to MD200 bit
Yes
Communication continued?
No
Writing 1 to ST2_0 bit
<6>
Clearing SAU2EN bit of PER1 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPS2 register after 4 or more clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-129 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

11.7.2 UART reception

UART reception is an operation wherein the 78K0R/Hx3 asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMR register of both the odd- and even-numbered channels must be set.

UART	UART2
Target channel	Channel 1 of SAU2
Pins used	RxD2
Interrupt	INTSR2, Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	• Framing error detection flag (FEF20) • Parity error detection flag (PEF20) • Overrun error detection flag (OVF20)
Transfer data length	7 to 9 bits or 16 bits
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR20 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps]Note
<R> Data phase	Forward input (default: high level) Reverse input (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 29 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-131. Example of Contents of Registers for UART Reception of UART2 (1/2)

(a) Serial output register 2 (SO2) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO2	0	0	0	0	0	0	CKO21 ×	CKO20 ×	0	0	0	0	0	0	SO21 ×	SO20 ×

<R> **(b) Serial output enable register 2 (SOE2)**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE2_1 0	SOE2_0 ×

<R> **(c) Serial channel start register 2 (SS2)**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS2_1 1	SS2_0 ×

(d) Serial mode register 21 (SMR21)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR21	CKS21 0/1	SCCS21 0	0	0	0	0	0	STS21 1	0	SIS210 0/1	1	0	0	MD212 0	MD211 1	MD210 0

0: Forward (normal) reception
1: Reverse reception

Interrupt sources of channel 1
0: Transfer end interrupt

(e) Serial mode register 20 (SMR20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR20	CKS20 0/1	SCCS20 0	0	0	0	0	0	STS20 0	0	SIS200 0	1	0	0	MD202 0	MD201 1	MD200 0/1

Same setting value as CKS21

Interrupt sources of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

(f) Serial communication operation setting register 21 (SCR21)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<R> SCR21	TXE21 0	RXE21 1	DAP21 0	CKP21 0	0	0	PTC211 0/1	PTC210 0/1	DIR21 0/1	0	SLC211 0	SLC210 1	DLS213 0/1	DLS212 0/1	DLS211 0/1	DLS210 0/1

Caution For the UART reception, be sure to set SMR20 of channel 0 that is to be paired with channel 1.

Remark □ : Setting is fixed in the UART reception mode, ▨ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-131. Example of Contents of Registers for UART Reception of UART2 (2/2)

(g) Serial data register 21 (SDR21)

(i) When operation is stopped (SE2_1 = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR21	Baud rate setting							0	0	0	0	0	0	0	0	0

(ii) During operation (SE2_1 = 1) (lower 8 bits: SDR21L)

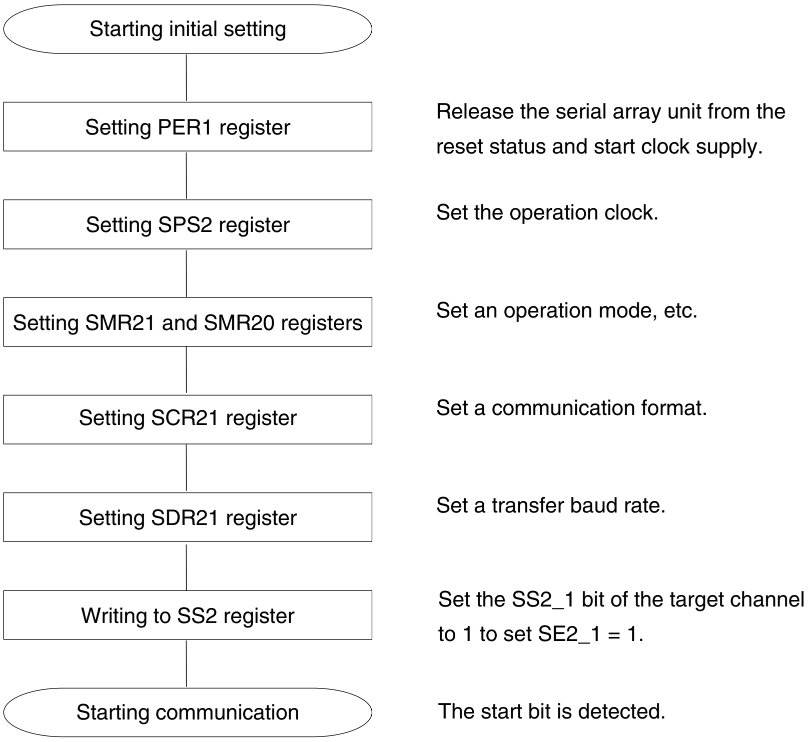
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR21	Transmit data setting															

SDR21L

Remark □: 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 11-132. Initial Setting Procedure for UART Reception


Starting initial setting
Setting PER1 register
Release the serial array unit from the reset status and start clock supply.
Setting SPS2 register
Set the operation clock.
Setting SMR21 and SMR20 registers
Set an operation mode, etc.
Setting SCR21 register
Set a communication format.
Setting SDR21 register
Set a transfer baud rate.
Writing to SS2 register
Set the SS2_1 bit of the target channel to 1 to set SE2_1 = 1.
Starting communication
The start bit is detected.

Caution After setting the PER1 register to 1, be sure to set the SPS2 register after 4 or more clocks have elapsed.

Figure 11-133. Procedure for Stopping UART Reception

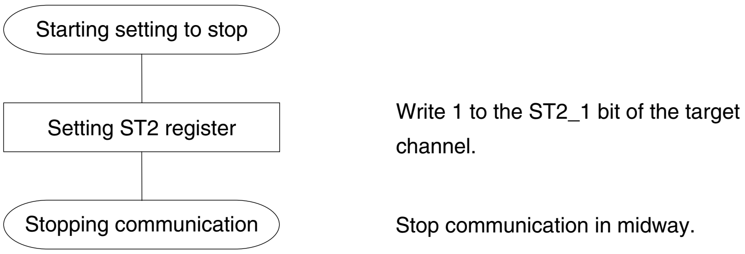
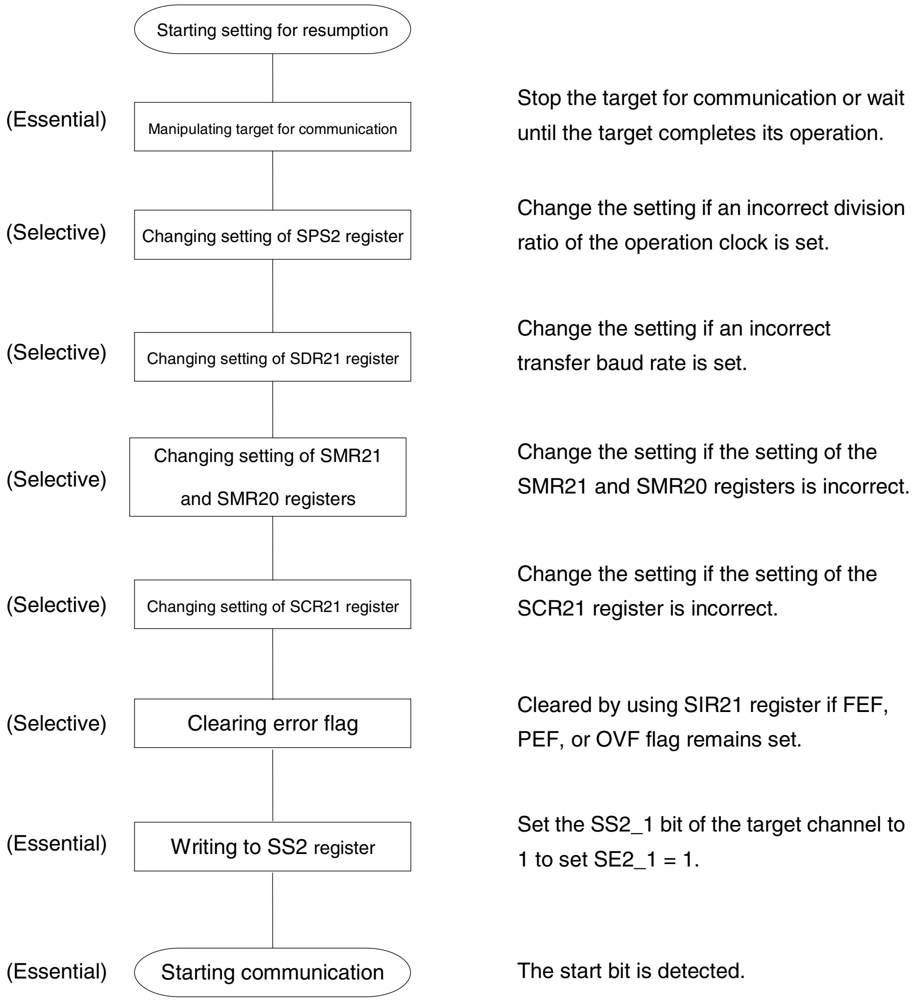

Starting setting to stop
Setting ST2 register
Write 1 to the ST2_1 bit of the target channel.
Stopping communication
Stop communication in midway.

Figure 11-134. Procedure for Resuming UART Reception


Starting setting for resumption
(Essential)
Manipulating target for communication
Stop the target for communication or wait until the target completes its operation.
(Selective)
Changing setting of SPS2 register
Change the setting if an incorrect division ratio of the operation clock is set.
(Selective)
Changing setting of SDR21 register
Change the setting if an incorrect transfer baud rate is set.
(Selective)
Changing setting of SMR21 and SMR20 registers
Change the setting if the setting of the SMR21 and SMR20 registers is incorrect.
(Selective)
Changing setting of SCR21 register
Change the setting if the setting of the SCR21 register is incorrect.
(Selective)
Clearing error flag
Cleared by using SIR21 register if FEF, PEF, or OVF flag remains set.
(Essential)
Writing to SS2 register
Set the SS2_1 bit of the target channel to 1 to set SE2_1 = 1.
(Essential)
Starting communication
The start bit is detected.

(3) Processing flow

Figure 11-135. Timing Chart of UART Reception

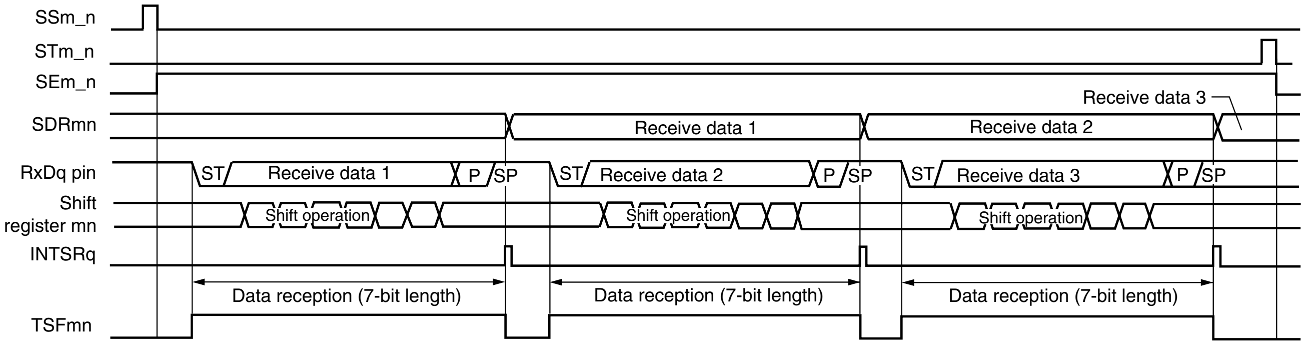
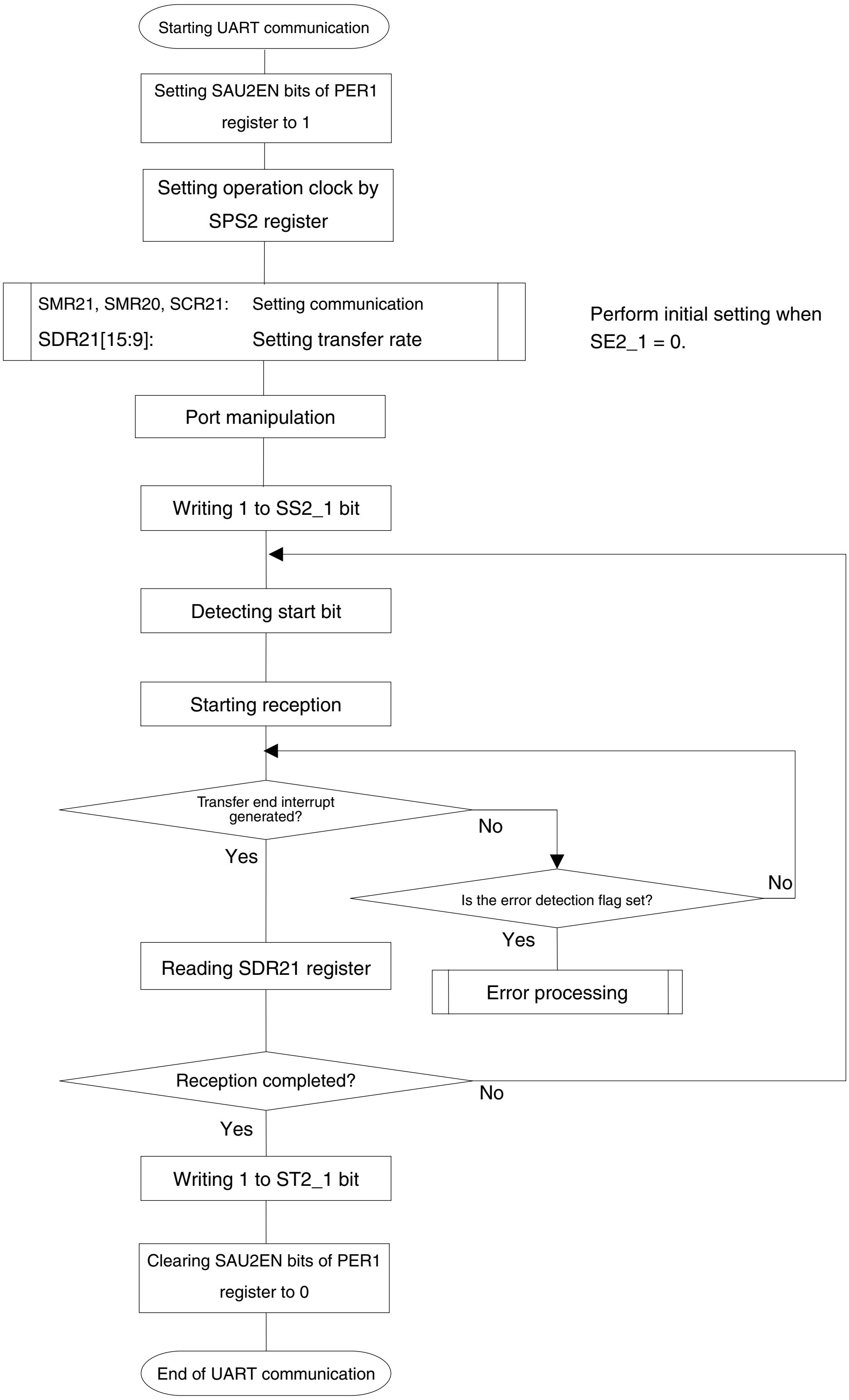


Figure 11-136. Flowchart of UART Reception


Starting UART communication
Setting SAU2EN bits of PER1 register to 1
Setting operation clock by SPS2 register
SMR21, SMR20, SCR21: Setting communication
SDR21[15:9]: Setting transfer rate
Perform initial setting when SE2_1 = 0.
Port manipulation
Writing 1 to SS2_1 bit
Detecting start bit
Starting reception
Transfer end interrupt generated?
No
Yes
Is the error detection flag set?
No
Yes
Reading SDR21 register
Error processing
Reception completed?
No
Yes
Writing 1 to ST2_1 bit
Clearing SAU2EN bits of PER1 register to 0
End of UART communication

Caution After setting the PER1 register to 1, be sure to set the SPS2 register after 4 or more clocks have elapsed.

11.7.3 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART2) communication can be calculated by the following expressions.

(Baud rate) [bps] = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDR2n[15:9] + 1) ÷ 2

Caution Setting SDR2n [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. n: Channel number (n = 0, 1)

The operation clock (f_{MCK}) is determined by serial clock select register 2 (SPS2) and bit 15 (CKS2n) of serial mode register 2n (SMR2n).

Table 11-6. Operating Clock Selection

SMR2n Register	SPS2 Register								Operation Clock (f_{MCK})[Note]	
CKS2n	PRS 213	PRS 212	PRS 211	PRS 210	PRS 203	PRS 202	PRS 201	PRS 200		f_{CLK} = 24 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	24 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	12 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^{2}$	6 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^{3}$	3 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^{4}$	1.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^{5}$	750 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^{6}$	375 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^{7}$	187.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^{8}$	93.75 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^{9}$	46.86 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	23.44 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	11.72 kHz
	X	X	X	X	1	1	1	1	INTTM23	
1	0	0	0	0	X	X	X	X	f_{CLK}	24 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	12 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^{2}$	6 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^{3}$	3 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^{4}$	1.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^{5}$	750 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^{6}$	375 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^{7}$	187.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^{8}$	93.75 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^{9}$	46.86 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	23.44 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	11.72 kHz
	1	1	1	1	X	X	X	X	INTTM23	
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (ST2 = 0003H) the operation of the serial array unit 2 (SAU2). When selecting INTTM23 for the operation clock, also stop the timer array unit 2 (TAU2) (TT2 = 00FFH).

Remarks 1. X: Don’t care

2. n: Channel number (n = 0, 1)

(2) Baud rate error during transmission

The baud rate error of UART (UART2) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) [%] = (Calculated baud rate value) ÷ (Target baud rate) × 100 − 100

Here is an example of setting a UART baud rate at f_{CLK} = 24 MHz.

<R>

UART Baud Rate (Target Baud Rate)	f_{CLK} = 24 MHz			
	Operation Clock (f_{MCK})	SDR2n[15:9]+1	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	78	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^8$	78	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^7$	78	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^6$	78	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^5$	78	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^4$	78	9615.38 bps	+0.16 %
10400 bps	$f_{CLK}/2^4$	72	10416.64 bps	+0.16 %
19200 bps	$f_{CLK}/2^3$	78	19230.8 bps	+0.16 %
31250 bps	$f_{CLK}/2^3$	48	31250.0 bps	±0.0 %
38400 bps	$f_{CLK}/2^2$	78	38461.5 bps	+0.16 %
76800 bps	$f_{CLK}/2$	78	76923.1 bps	+0.16 %
153600 bps	f_{CLK}	78	153846 bps	+0.16 %
312500 bps	f_{CLK}	38	315781.25 bps	+1.05 %

<R>

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART2) 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.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

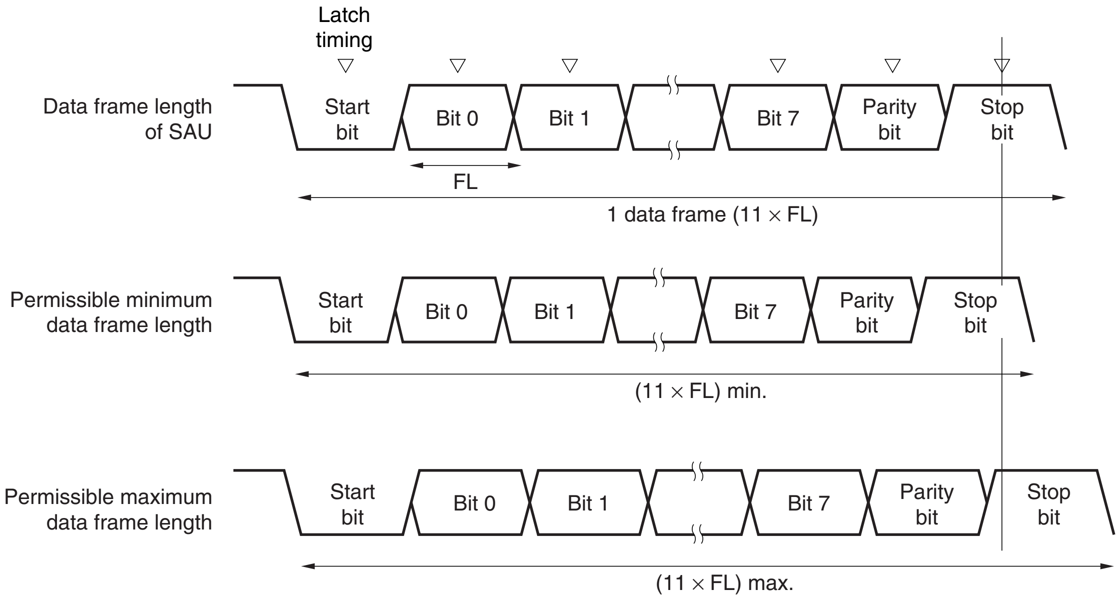
$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **11.7.3 (1) Baud rate calculation expression**.)

k: SDR21[15:9] + 1

Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Figure 11-137. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 11-137, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of the serial data register 21 (SDR21) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

11.7.4 Procedure for processing errors that occurred during UART communication

The procedure for processing errors that occurred during UART communication is described in Figures 11-138 and 11-139.

Figure 11-138. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads SDR2n register. →	The BFF0 = 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 SSR2n register.		Error type is identified and the read value is used to clear error flag.
Writes SIR2n register. →	Error flag is cleared.	Only error generated at the point of reading can be cleared, by writing the value read from the SSR2n register to the SIR2n register without modification.

Figure 11-139. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads SDR2n register. →	The 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 SSR2n register.		Error type is identified and the read value is used to clear error flag.
Writes SIR2n register. →	Error flag is cleared.	Only error generated at the point of reading can be cleared, by writing the value read from the SSR2n register to the SIR2n register without modification.
Sets ST2_n bit to 1. →	The SE2_n = 0, and channel n stops operating.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets SS2_n bit to 1. →	The SE2_n = 1, and channel n is enabled to operate.	

Remark n: Channel number (n = 0, 1)

11.8 Operation of Simplified I^{2}C (IIC11, 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]

- Overrun error
- Parity error (ACK error)

* [Functions not supported by simplified I^{2}C]

- 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 SOEm_n (SOEm register) bit and stopping the output of serial communication data. See **11.8.3 (2) Processing flow** for details.

Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1)

The channels supporting simplified I^{2}C (IIC11, IIC20) are channel 1 of SAU1 and channel 0 of SAU2.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^{2}C
0	0	CSI00 (supports SPI)	–	–
	1	CSI01 (supports SPI)		–
1	0	CSI20	–	–
	1	CSI21		IIC11
2	0	–	UART2	IIC20
	1	–		–

Simplified I^{2}C (IIC11, IIC20) performs the following four types of communication operations.

- Address field transmission (See **11.8.1**.)
- Data transmission (See **11.8.2**.)
- Data reception (See **11.8.3**.)
- Stop condition generation (See **11.8.4**.)

11.8.1 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	IIC11	IIC20
Target channel	Channel 1 of SAU1	Channel 0 of SAU2
Pins used	SCL11, SDA11	SCL20, SDA20^Note
Interrupt	INTIIC11	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Parity error detection flag (PEF11)	Parity error detection flag (PEF20)
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)	
Transfer rate	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 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 N-ch open-drain output (V_{DD} tolerance) mode (POM4_3 = 1) for the port output mode registers (POM4) (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 (POM4_2 = 1) also for the clock input/output pins (SCL20) (see **4.4.4 Connecting to external device with different potential (3 V)** for details).

(1) Register setting

Figure 11-140. Example of Contents of Registers for Address Field Transmission of Simplified I^{2}C (IIC11, IIC20) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

Start condition is generated by manipulating the SOmn bit.

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm_1 0/1	SOEm_0 0/1

SOEm_n = 0 until the start condition is generated, and

SOEm_n = 1 after generation.

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm_1 0/1	SSm_0 0/1

(d) 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	SCCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

Interrupt sources of channel n
0: Transfer end interrupt

(e) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0	SLCmn0 1	0	DLSmn2 1	DLSmn1 1	DLSmn0 1

Setting of parity bit
00B: No parity

Setting of stop bit
01B: Appending 1 bit

(f) Serial data register mn (SDRmn)

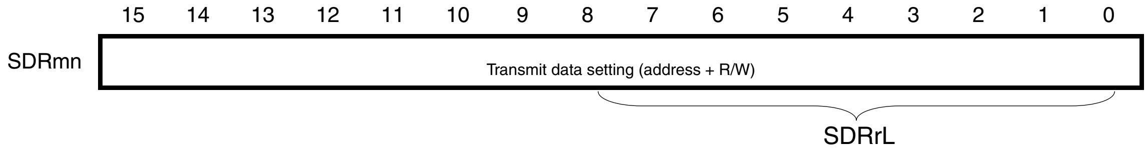
(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	0	0	0	0	0	0	0	0

Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1), r: IIC number (r = 11, 20)
□ : Setting is fixed in the IIC mode, ▨ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-140. Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC11, IIC20) (2/2)

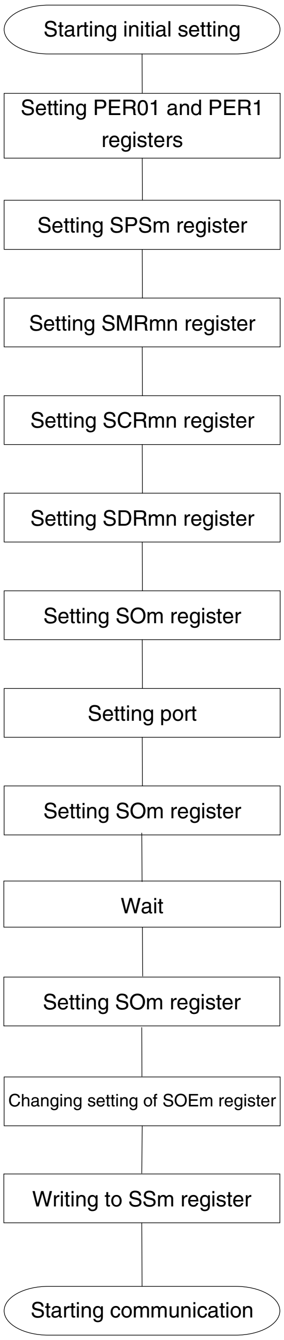
(ii) During operation (SEm_n = 1) (lower 8 bits: SDRrL)



Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1), r: IIC number (r = 11, 20)
□ : Setting is fixed in the IIC mode, ▨ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-141. Initial Setting Procedure for Address Field Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate.

Manipulate the SOmn and CKO0mn bits and set an initial output level.

Enable data output, clock output, and the N-ch open-drain output (V_{DD} tolerance) mode of the target channel by setting a port register, a port mode register, and a port output mode register.

Clear the SOmn bit to 0 to generate the start condition.

Secure a wait time so that the specifications of I^2C on the slave side are satisfied.

Clear the CKOmn bit to 0 to lower the clock output level.

Set the SOEm_n bit to 1 and enable data output of the target channel.

Set the SSm_n bit of the target channel to 1 to set SEm_n = 1.

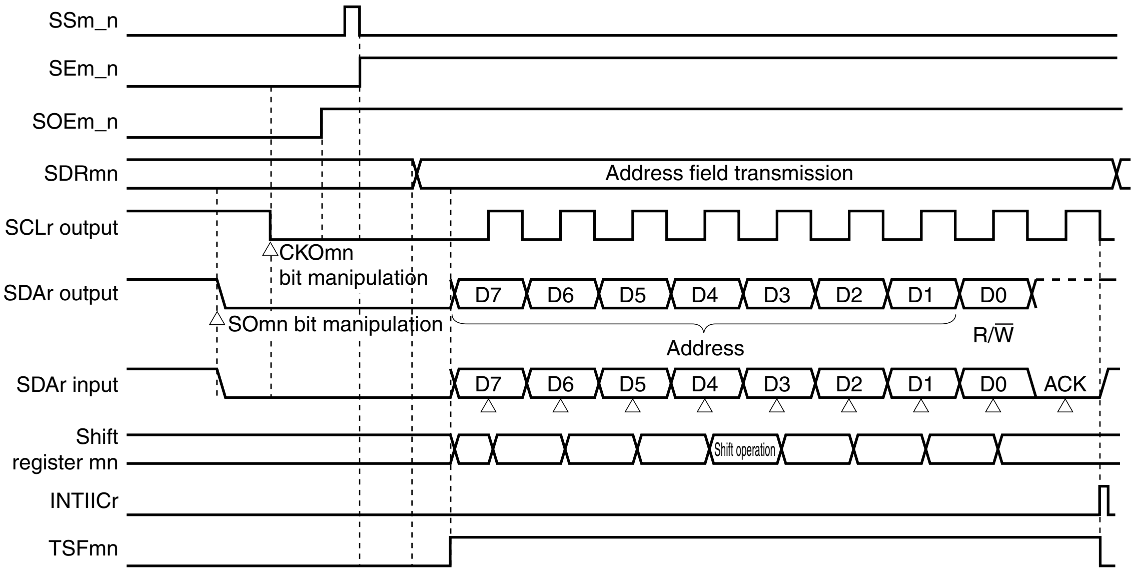
Set address and R/W to the SDRmn register and start communication.

Caution After setting the PER0 and PER1 registers to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1), r: IIC number (r = 11, 20)

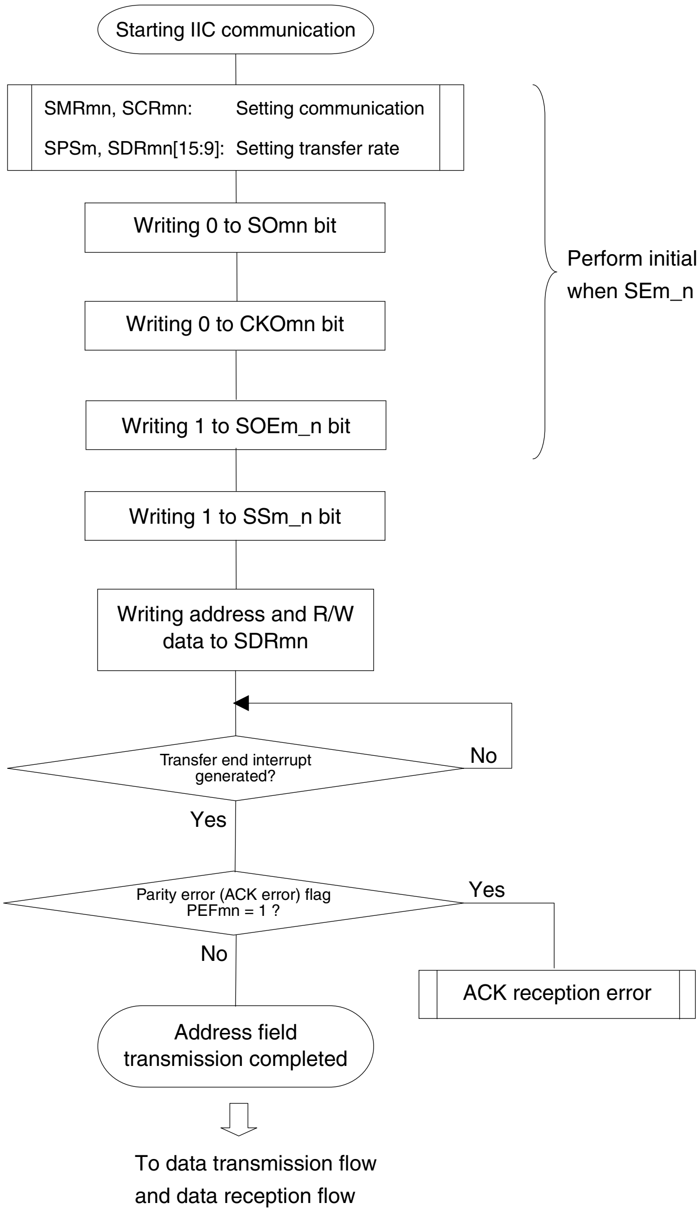
(3) Processing flow

Figure 11-142. Timing Chart of Address Field Transmission



Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1), r: IIC number (r = 11, 20)

Figure 11-143. Flowchart of Address Field Transmission


Starting IIC communication
SMRmn, SCRmn: Setting communication
SPSm, SDRmn[15:9]: Setting transfer rate
Writing 0 to SOmn bit
Writing 0 to CKOmn bit
Writing 1 to SOEm_n bit
Perform initial setting when SEm_n = 0.
Writing 1 to SSm_n bit
Writing address and R/W data to SDRmn
Transfer end interrupt generated?
No
Yes
Parity error (ACK error) flag PEFmn = 1 ?
Yes
No
ACK reception error
Address field transmission completed
To data transmission flow and data reception flow

Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1)

11.8.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC11	IIC20
Target channel	Channel 1 of SAU1	Channel 0 of SAU2
Pins used	SCL11, SDA11	SCL20, SDA20[Note]
Interrupt	INTIIC11	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Parity error detection flag (PEF11)	Parity error detection flag (PEF20)
Transfer data length	8 bits	
Transfer rate	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^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 N-ch open-drain output (V_{DD} tolerance) mode (POM4_3 = 1) for the port output mode registers (POM4) (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 (POM4_2 = 1) also for the clock input/output pins (SCL20) (see **4.4.4 Connecting to external device with different potential (3 V)** for details).

(1) Register setting

Figure 11-144. Example of Contents of Registers for Data Transmission of Simplified I^{2}C (IIC11, IIC20) (1/2)

(a) Serial output register m (SOm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	×	0/1Note	0	0	0	0	0	0	×	0/1Note

(b) Serial output enable register m (SOEm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm															SOEm_1	SOEm_0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

(c) Serial channel start register m (SSm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm															SSm_1	SSm_0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

(d) Serial mode register mn (SMRmn) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	SCCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(e) Serial communication operation setting register mn (SCRmn) … Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn			PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0		DLSmn2	DLSmn1	DLSmn0
	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

Note The value varies depending on the communication data during communication operation.

Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1), r: IIC number (r = 11, 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

Figure 11-144. Example of Contents of Registers for Data Transmission of Simplified I^{2}C (IIC11, IIC20) (2/2)

(f) Serial data register mn (SDRmn)

(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	0	0	0	0	0	0	0	0

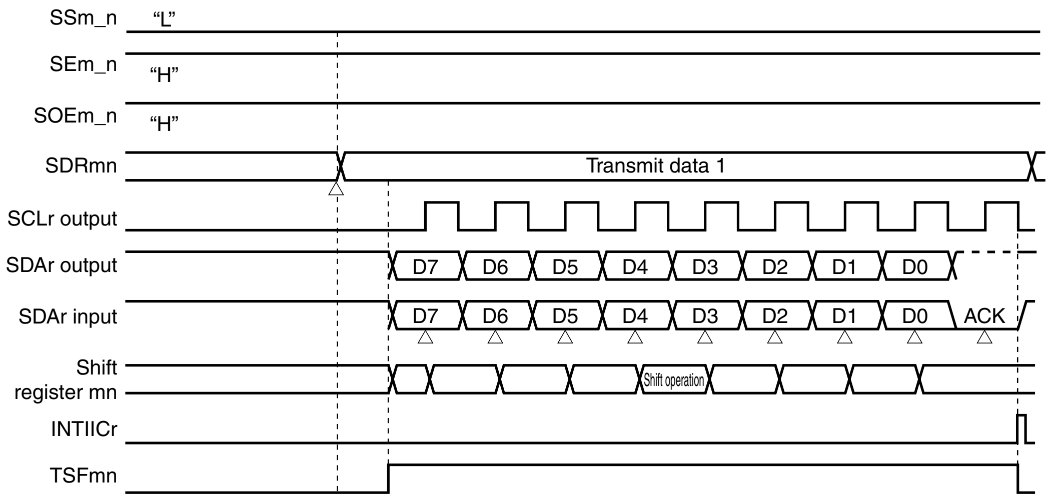
(ii) During operation (SEm_n = 1) (lower 8 bits: SDRrL)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Transmit data setting															
									SDRrL							

Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1), r: IIC number (r = 11, 20)
□: Setting is fixed in the IIC mode, ▨: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

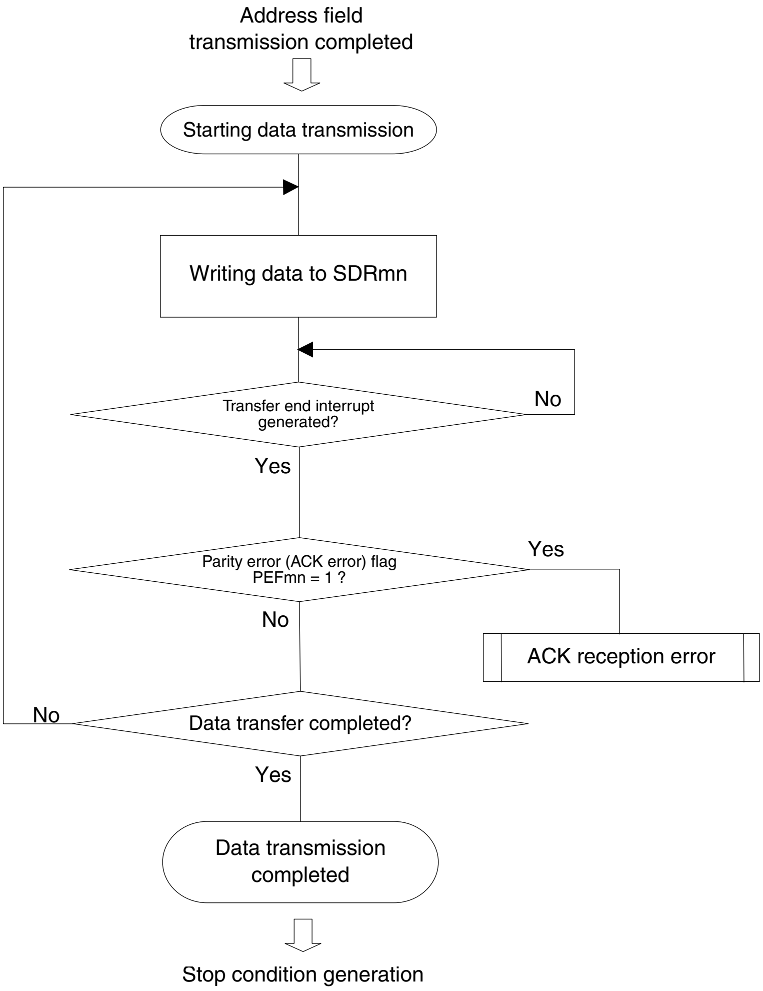
(2) Processing flow

Figure 11-145. Timing Chart of Data Transmission



Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1), r: IIC number (r = 11, 20)

Figure 11-146. Flowchart of Data Transmission


Address field transmission completed
Starting data transmission
Writing data to SDRmn
Transfer end interrupt generated?
No
Yes
Parity error (ACK error) flag PEFmn = 1 ?
Yes
No
ACK reception error
No
Data transfer completed?
Yes
Data transmission completed
Stop condition generation

11.8.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC11	IIC20
Target channel	Channel 1 of SAU1	Channel 0 of SAU2
Pins used	SCL11, SDA11	SCL20, SDA20[Note]
Interrupt	INTIIC11	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	8 bits	
Transfer rate	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)	
<R> Data level	Forward input (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 N-ch open-drain output (V_{DD} tolerance) mode (POM4_3 = 1) for the port output mode registers (POM4) (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 (POM4_2 = 1) also for the clock input/output pins (SCL20) (see **4.4.4 Connecting to external device with different potential (3 V)** for details).

(1) Register setting

Figure 11-147. Example of Contents of Registers for Data Reception of Simplified I^{2}C (IIC11, IIC20) (1/2)

(a) Serial output register m (SOm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 0/1[Note]	0	0	0	0	0	0	SOm1 ×	SOm0 0/1[Note]

(b) Serial output enable register m (SOEm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm_1 0/1	SOEm_0 0/1

(c) Serial channel start register m (SSm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm_1 0/1	SSm_0 0/1

(d) Serial mode register mn (SMRmn) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	SCCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(e) Serial communication operation setting register mn (SCRmn) … Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0	SLCmn0 1	0	DLSmn2 1	DLSmn1 1	DLSmn0 1

Note The value varies depending on the communication data during communication operation.

Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1), r: IIC number (r = 11, 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

Figure 11-147. Example of Contents of Registers for Data Reception of Simplified I^{2}C (IIC11, IIC20) (2/2)

(f) Serial data register mn (SDRmn)

(i) When operation is stopped (SEm_n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	0	0	0	0	0	0	0	0

(ii) During operation (SEm_n = 1) (lower 8 bits: SDRrL)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Dummy transmit data setting (FFH)															
									SDRrL							

Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1), r: IIC number (r = 11, 20)

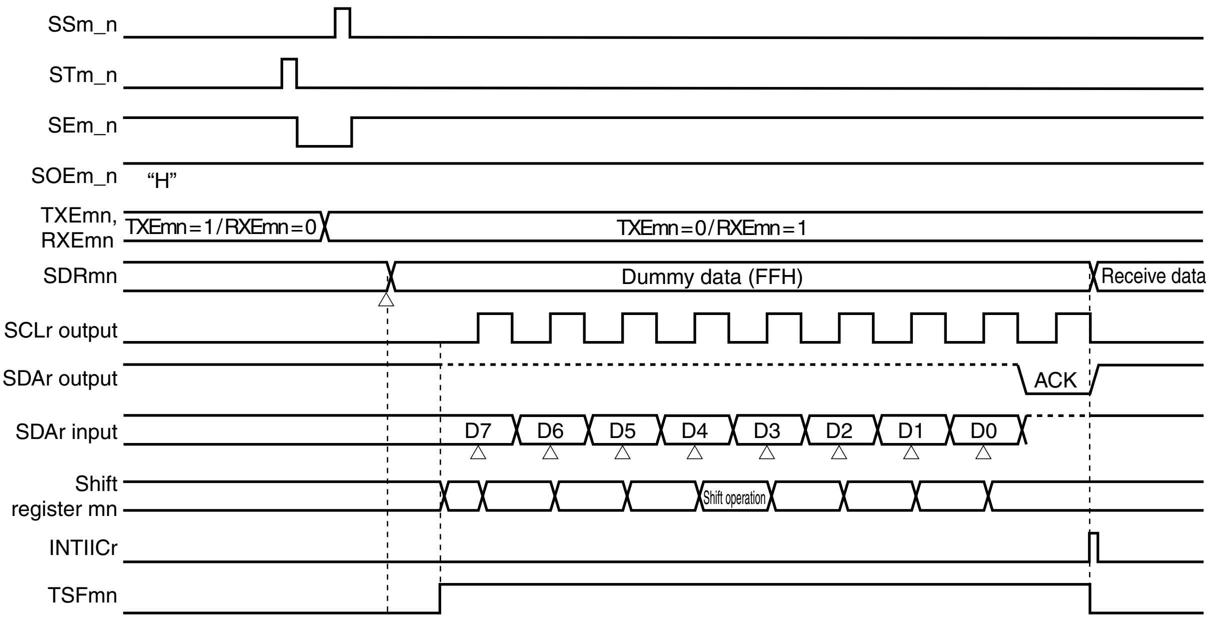
□ : Setting is fixed in the IIC mode, ▨ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

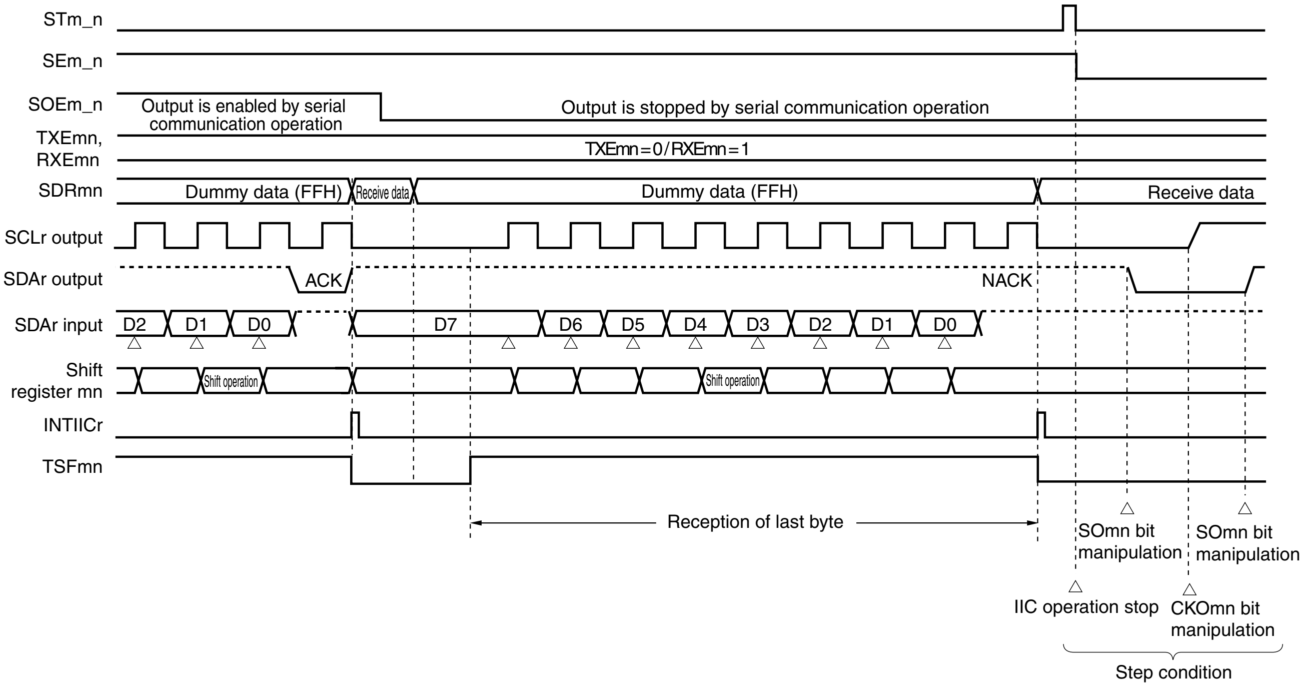
(2) Processing flow

Figure 11-148. Timing Chart of Data Reception

(a) When starting data reception

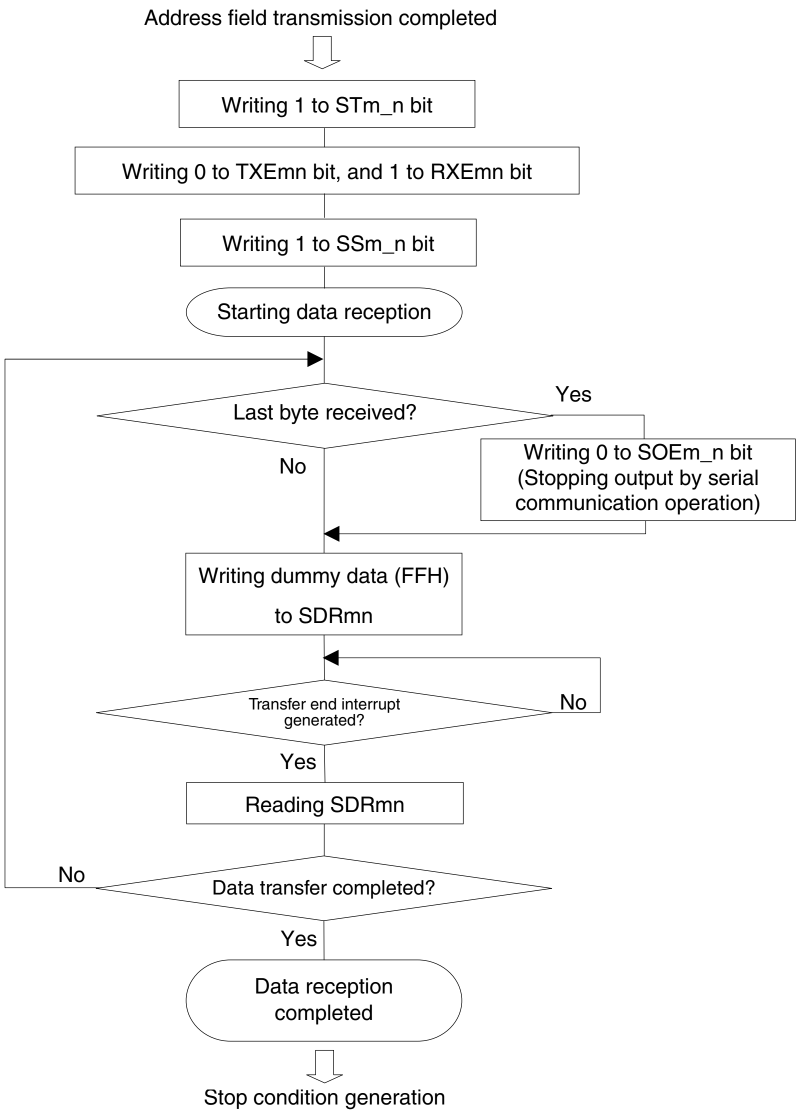


(b) When receiving last data



Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1), r: IIC number (r = 11, 20)

Figure 11-149. Flowchart of Data Reception



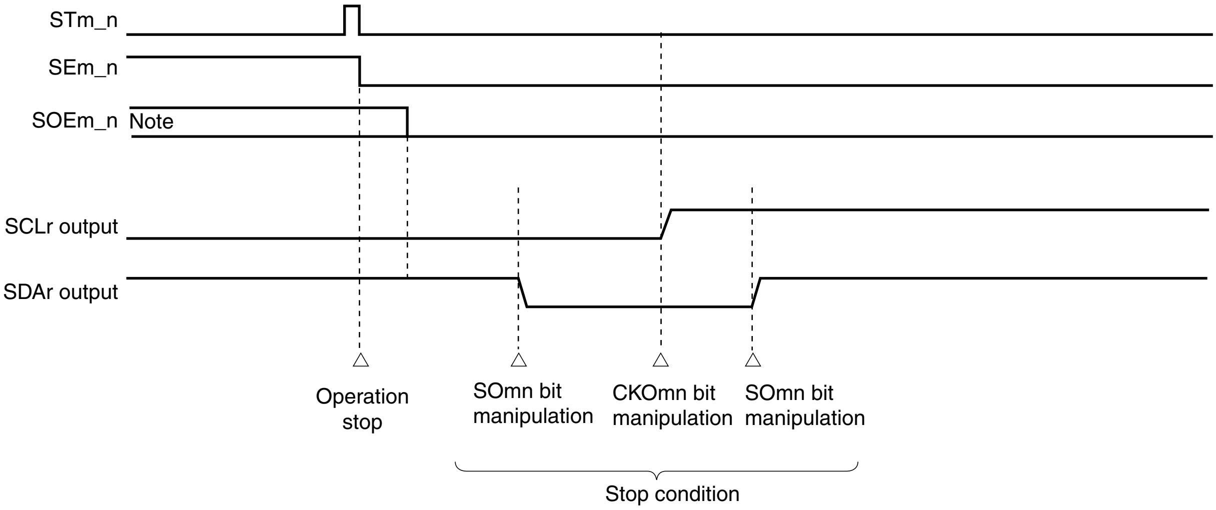
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting "1" to the STm_n bit to stop operation and generating a stop condition.

11.8.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

(1) Processing flow

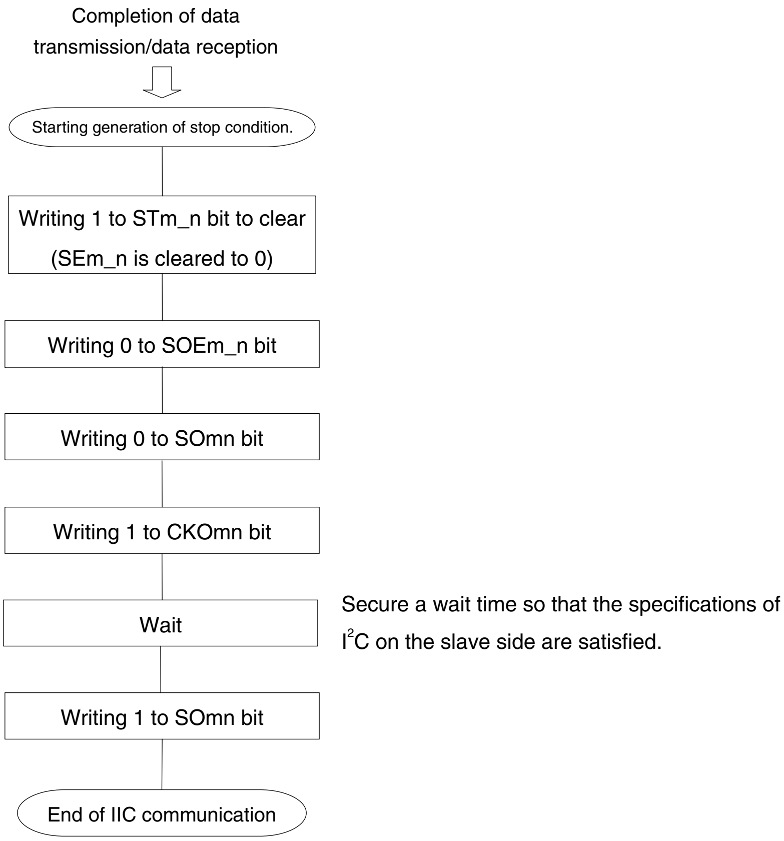
Figure 11-150. Timing Chart of Stop Condition Generation



Note During the receive operation, the SOEm_n bit is set to 0 before receiving the last data.

Remark m: Unit number (m = 1, 2), n: Channel number (n = 0, 1), r: IIC number (r = 11, 20)

Figure 11-151. Flowchart of Stop Condition Generation


Completion of data transmission/data reception
Starting generation of stop condition.
Writing 1 to STm_n bit to clear (SEm_n is cleared to 0)
Writing 0 to SOEm_n bit
Writing 0 to SOmn bit
Writing 1 to CKOmn bit
Wait
Secure a wait time so that the specifications of I2C on the slave side are satisfied.
Writing 1 to SOmn bit
End of IIC communication

11.8.5 Calculating transfer rate

The transfer rate for simplified I^{2}C (IIC11, IIC20) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

Caution Setting SDRmn[15:9] = 0000000B is prohibited. Setting SDRmn[15:9] = 0000001B or more.

Remarks 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000001B to 1111111B) and therefore is 1 to 127.

2. m: Unit number (m = 1, 2), n: Channel number (n = 0, 1)

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11-7. Operating Clock Selection

SMRmn Register	SPSm Register								Operation Clock (f_{MCK})[Note]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	24 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	12 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	93.75 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	46.86 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	23.44 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	11.72 kHz
	X	X	X	X	1	1	1	1	INTTM23	
1	0	0	0	0	X	X	X	X	f_{CLK}	24 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	12 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	93.75 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	46.86 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	23.44 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	11.72 kHz
	1	1	1	1	X	X	X	X	INTTM23	
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (ST2 = 0003H) the operation of the serial array unit m (SAUm). When selecting INTTM23 for the operation clock, also stop the timer array unit 2 (TAU2) (TT2 = 00FFH).

Remarks 1. X: Don’t care

2. m: Unit number (m = 1, 2), n: Channel number (n = 0, 1)

Here is an example of setting an IIC transfer rate where f_{MCK} = f_{CLK} = 24 MHz.

<R>

IIC Transfer Mode (Desired Transfer Rate)	f_{CLK} = 24 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9] +1	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	f_{CLK}	120	100 kHz	0.0%
400 kHz	f_{CLK}	30	400 kHz	0.0%

11.9 Procedure for processing errors that occurred during simplified I^{2}C (IIC11, IIC20) communication

The procedure for processing errors that occurred during simplified I^{2}C (IIC11, IIC20) communication is described in Figures 11-152 and 11-153.

Figure 11-152. 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.	Only error generated at the point of reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 11-153. Processing Procedure in Case of Parity Error (ACK error) in Simplified I^{2}C 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.	Only error generated at the point of reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets STm_n bit to 1. →	SEm_n = 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 SSm_n bit to 1. →	SEm_n = 1, and channel n is enabled to operate.	

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 1)

11.10 Relationship Between Register Settings and Pins

Tables 11-8 to 11-14 show the relationship between register settings and pins for each channel of serial array units 0 to 2.

Table 11-8. Relationship Between Register Settings and Pins (Channel 0 of Unit 0: CSI00, STSCSI00 = 0)

SE 0_0[Note 1]	MD 002	MD 001	SOE 0_0	SO 00	CKO 00	TXE 00	RXE 00	SSE 0_0	PM 1_7	P1_7	PM 1_6	P1_6	PM 1_5	P1_5	PM 3_0	P3_0	Operation Mode	Pin Function			
																		P17/SCK00/TI14/TO14	P16/SI00/TI12/TO12	P15/SO00/TI10/TO10	P30/SSI00/INTP2/TI01/TO01
0	0	0	0	1	1	0	0	0	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	Operation stop mode	P17/TI14/TO14	P16/TI12/TO12	P15/TO10/TO10	P30/INTP2/TI01/TO01
1	0	0	0	1	1	0	1	0	1	×	1	×	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	Slave CSI00 reception	SCK00 (input)	SI00	P15	P30
			0	1	1	0	1	1	1	×	1	×	×[Note 2]	×[Note 2]	1	×	Slave CSI00 reception (SPI)	SCK00 (input)	SI00	P15	SSI00
			1	0/1[Note 3]	1	1	0	0	1	×	×	×	0	1	×[Note 2]	×[Note 2]	Slave CSI00 transmission	SCK00 (input)	P16	SO00	P30
			1	0/1[Note 3]	1	1	0	1	1	×	×	×	0	1	1	×	Slave CSI00 transmission (SPI)	SCK00 (input)	P16	SO00	SSI00
			1	0/1[Note 3]	1	1	1	0	1	×	1	×	0	1	×[Note 2]	×[Note 2]	Slave CSI00 transmission/reception	SCK00 (input)	SI00	SO00	P30
			1	0/1[Note 3]	1	1	1	1	1	×	1	×	0	1	1	×	Slave CSI00 transmission/reception (SPI)	SCK00 (input)	SI00	SO00	SSI00
			0	1	0/1	0	1	×	0	1	1	×	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	Master CSI00 reception	SCK00 (output)	SI00	P15	P30
			1	0/1[Note 3]	0/1[Note 3]	1	0	×	0	1	×	×	0	1	×[Note 2]	×[Note 2]	Master CSI00 transmission	SCK00 (output)	P16	SO00	P30
			1	0/1[Note 3]	0/1[Note 3]	1	1	×	0	1	1	×	0	1	×[Note 2]	×[Note 2]	Master CSI00 transmission/reception	SCK00 (output)	SI00	SO00	P30

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 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, see **11.3 (12) Serial output register m (SOm)**.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark X: Don't care

Table 11-9. Relationship Between Register Settings and Pins (Channel 0 of Unit 0: CSI00, STSCSI00 = 1)

SE 0_0 Note 1	MD 002	MD 001	SOE 0_0	SO 00	CKO 00	TXE 00	RXE 00	SSE 0_0	PM 6_0	P6_0	PM 6_1	P6_1	PM 6_2	P6_2	PM 6_3	P6_3	Operation Mode	Pin Function			
																		P60/SCK00/ SCL11	P61/SI00/ SDA11	P62/SO00	P63/SSI00
0	0	0	0	1	1	0	0	0	×Note 2	×Note 2	×Note 2	×Note 2	×Note 2	×Note 2	×Note 2	×Note 2	Operation stop mode	P60	P61	P62	P63
1	0	0	0	1	1	0	1	0	1	×	1	×	×Note 2	×Note 2	×Note 2	×Note 2	Slave CSI00 reception	SCK00 (input)	SI00	P62	P63
			0	1	1	0	1	1	1	×	1	×	×Note 2	×Note 2	1	×	Slave CSI00 reception (SPI)	SCK00 (input)	SI00	P62	SSI00
			1	0/1 Note 3	1	1	0	0	1	×	×Note 2	×Note 2	0	1	×Note 2	×Note 2	Slave CSI00 transmission	SCK00 (input)	P61	SO00	P63
			1	0/1 Note 3	1	1	0	1	1	×	×Note 2	×Note 2	0	1	1	×	Slave CSI00 transmission (SPI)	SCK00 (input)	P61	SO00	SSI00
			1	0/1 Note 3	1	1	1	0	1	×	1	×	0	1	×Note 2	×Note 2	Slave CSI00 transmission/ reception	SCK00 (input)	SI00	SO00	P30
			1	0/1 Note 3	1	1	1	1	1	×	1	×	0	1	1	×	Slave CSI00 transmission/ reception (SPI)	SCK00 (input)	SI00	SO00	SSI00
			0	1	0/1	0	1	×	0	1	1	×	×Note 2	×Note 2	×Note 2	×Note 2	Master CSI00 reception	SCK00 (output)	SI00	P62	P63
			1	0/1 Note 3	0/1 Note 3	1	0	×	0	1	×Note 2	×Note 2	0	1	×Note 2	×Note 2	Master CSI00 transmission	SCK00 (output)	P61	SO00	P63
			1	0/1 Note 3	0/1 Note 3	1	1	×	0	1	1	×	0	1	×Note 2	×Note 2	Master CSI00 transmission/ reception	SCK00 (output)	SI00	SO00	P63

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 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, see **11.3 (12) Serial output register m (SOm)**.

Remark X: Don't care

Table 11-10. Relationship Between Register Settings and Pins (Channel 1 of Unit 0: CSI01)

SE 0_1 Note 1	MD 012	MD 011	SOE 0_1	SO 01	CKO 01	TXE 01	RXE 01	SSE 0_1	PM 7_6	P7_6	PM 7_5	P7_5	PM 7_4	P7_4	PM 7_7	P7_7	Operation Mode	Pin Function			
																		P76/$\overline{\text{SCK01}}$/ KR6	P75/SI01/ KR5	P74/SO01/ KR4	P77/$\overline{\text{SSI01}}$/ KR7
0	0	0	0	1	1	0	0	0	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	Operation stop mode	P76	P75	P74	P77
1	0	0	0	1	1	0	1	0	1	×	1	×	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	Slave CSI01 reception	$\overline{\text{SCK01}}$ (input)	SI01	P74	P77
			0	1	1	0	1	1	1	×	1	×	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	1	×	Slave CSI01 reception (SPI)	$\overline{\text{SCK01}}$ (input)	SI01	P74	$\overline{\text{SSI01}}$
			1	0/1 Note 3	1	1	0	0	1	×	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	0	1	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	Slave CSI01 transmission	$\overline{\text{SCK01}}$ (input)	P75	SO01	P77
			1	0/1 Note 3	1	1	0	1	1	×	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	0	1	1	×	Slave CSI01 transmission (SPI)	$\overline{\text{SCK01}}$ (input)	P75	SO01	$\overline{\text{SSI01}}$
			1	0/1 Note 3	1	1	1	0	1	×	1	×	0	1	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	Slave CSI01 transmission/ reception	$\overline{\text{SCK01}}$ (input)	SI01	SO01	P77
			1	0/1 Note 3	1	1	1	1	1	×	1	×	0	1	1	×	Slave CSI01 transmission/ reception (SPI)	$\overline{\text{SCK01}}$ (input)	SI01	SO01	$\overline{\text{SSI01}}$
			0	1	0/1	0	1	×	0	1	1	×	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	Master CSI01 reception	$\overline{\text{SCK01}}$ (output)	SI01	P74	P77
			1	0/1 Note 3	0/1 Note 3	1	0	×	0	1	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	0	1	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	Master CSI01 transmission	$\overline{\text{SCK01}}$ (output)	P75	SO01	P77
			1	0/1 Note 3	0/1 Note 3	1	1	×	0	1	1	×	0	1	$\times^{\text{Note 2}}$	$\times^{\text{Note 2}}$	Master CSI01 transmission/ reception	$\overline{\text{SCK01}}$ (output)	SI01	SO01	P77

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 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, see **11.3 (12) Serial output register m (SOm)**.

Remark X: Don’t care

Table 11-11. Relationship Between Register Settings and Pins (Channel 0 of Unit 1: CSI10)

SE 1_0 Note 1	MD 102	MD 101	SOE 1_0	SO 10	CKO 10	TXE 10	RXE 10	PM 1_0	P1_0	PM 1_1	P1_1	PM 1_2	P1_2	Operation Mode	Pin Function		
															P10/SCK10/ CTxD/LTxD1/ TI00/TO00	P11/SI10/ CRxD/LRxD1/ INTPLR1/TI02/ TO02	P12/SO10/ INTP3/TI16/ TO16
0	0	0	0	1	1	0	0	×Note 2	×Note 2	×Note 2	×Note 2	×Note 2	×Note 2	Operation stop mode	P10/CTxD/ LTxD1/TI00/ TO00	P11/CRxD/ LRxD1/ INTPLR1/TI02/ TO02	P12/INTP3/TI16/ TO16
1	0	0	0	1	1	0	1	1	×	1	×	×Note 2	×Note 2	Slave CSI10 reception	SCK10 (input)	SI10	P12
			1	0/1 Note 3	1	1	1	1	×	×Note 2	×Note 2	0	1	Slave CSI10 transmission	SCK10 (input)	P11	SO10
			1	0/1 Note 3	1	1	1	1	×	1	×	0	1	Slave CSI10 transmission/ reception	SCK10 (input)	SI10	SO10
			0	1	0/1	0	1	0	1	1	×	×Note 2	×Note 2	Master CSI10 reception	SCK10 (output)	SI10	P12
			1	0/1 Note 3	0/1 Note 3	1	0	0	1	×Note 2	×Note 2	0	1	Master CSI10 transmission	SCK10 (output)	P11	SO10
			1	0/1 Note 3	0/1 Note 3	1	1	0	1	1	×	0	1	Master CSI10 transmission/ reception	SCK10 (output)	SI10	SO10

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, see **11.3 (12) Serial output register m (SOm)**.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark X: Don't care

Table 11-12. Relationship Between Register Settings and Pins (Channel 1 of Unit 1: CSI11, IIC11) (1/2)

SE 1_1[Note 1]	MD 112	MD 111	SOE 1_1	SO 11	CKO 11	TXE 11	RXE 11	PM 15_3	P 15_3	PM 15_2	P 15_2	PM 15_1	P 15_1	PM 6_0	P 6_0	PM 6_1	P 6_1	Operation Mode	Pin Function				
																			P153/$\overline{\text{SCK11}}$	P152/SI11	P151/SO11	P60/SCK00/SCL11	P61/SI00/SDA11
0	0	0	0	1	1	0	0	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	Operation stop mode	P153	P152	P151	P60	P61
	1	0																					
1	0	0	0	1	1	0	1	1	×	1	×	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	Slave CSI11 reception	$\overline{\text{SCK11}}$ (input)	SI11	P151	P60	P61
			1	0/1[Note 3]	1	1	0	1	×	×[Note 2]	×[Note 2]	0	1	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	Slave CSI11 transmission	$\overline{\text{SCK11}}$ (input)	P152	SO11	P60	P61
			1	0/1[Note 3]	1	1	1	1	×	1	×	0	1	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	Slave CSI11 transmission/ reception	$\overline{\text{SCK11}}$ (input)	SI11	SO11	P60	P61
			0	1	0/1	0	1	0	1	1	×	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	Master CSI11 reception	$\overline{\text{SCK11}}$ (output)	SI11	P151	P60	P61
			1	0/1[Note 3]	0/1[Note 3]	1	0	0	1	×[Note 2]	×[Note 2]	0	1	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	Master CSI11 transmission	$\overline{\text{SCK11}}$ (output)	P152	SO11	P60	P61
			1	0/1[Note 3]	0/1[Note 3]	1	1	0	1	1	×	0	1	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	Master CSI11 transmission/ reception	$\overline{\text{SCK11}}$ (output)	SI11	SO11	P60	P61

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 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, see **11.3 (12) Serial output register m (SOm)**.

Remark X: Don’t care

Table 11-12. Relationship Between Register Settings and Pins (Channel 1 of Unit 1: CSI11, IIC11) (2/2)

SE	MD	MD	SOE	SO	CKO	TXE	RXE	PM	P	PM	P	PM	P	PM	P	PM	P	Operation	Pin Function				
1_1 Note 1	112	111	1_1	11	11	11	11	15_3	15_3	15_2	15_2	15_1	15_1	6_0	6_0	6_1	6_1	Mode	P153/ SCK11	P152/SI11	P151/ SO11	P60/ SCK00/ SCL11	P61/SI00/ SDA11
0	1	0	0	0/1 Note 2	0/1 Note 2	0	0	×Note 3	×Note 3	×Note 3	×Note 3	×Note 3	×Note 3	0	1	0	1	IIC11 start condition	P153	P152	P151	SCL11	SDA11
						1	0																
						0	1																
1			1	0/1 Note 4	0/1 Note 4	1	0	×Note 3	×Note 3	×Note 3	×Note 3	×Note 3	×Note 3	0	1	0	1	IIC11 address field transmission	P153	P152	P151	SCL11	SDA11
			1	0/1 Note 4	0/1 Note 4	1	0	×Note 3	×Note 3	×Note 3	×Note 3	×Note 3	×Note 3	0	1	0	1	IIC11 data transmission	P153	P152	P151	SCL11	SDA11
			1	0/1 Note 4	0/1 Note 4	0	1	×Note 3	×Note 3	×Note 3	×Note 3	×Note 3	×Note 3	0	1	0	1	IIC11 data reception	P153	P152	P151	SCL11	SDA11
0			0	0/1 Note 5	0/1 Note 5	0	1	×Note 3	×Note 3	×Note 3	×Note 3	×Note 3	×Note 3	0	1	0	1	IIC11 stop condition	P153	P152	P151	SCL11	SDA11
						1	0																
						0	1																

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. Set the CKO11 bit to 1 before a start condition is generated. Clear the SO11 bit from 1 to 0 when the start condition is generated.

3. This pin can be set as a port function pin.

4. This is 0 or 1, depending on the communication operation. For details, see **11.3 (12) Serial output register m (SOm)**.

5. Set the CKO11 bit to 1 before a stop condition is generated. Clear the SO11 bit from 0 to 1 when the stop condition is generated.

Remark X: Don’t care

Table 11-13. Relationship Between Register Settings and Pins (Channel 0 of Unit 2: UART2 Transmission, IIC20)

SE 2_0 [Note 1]	MD 202	MD 201	SOE 2_0	SO 20	CKO 20	TXE 20	RXE 20	PM 4_2	P4_2	PM 4_3	P4_3	Operation Mode	Pin Function	
													P42/TxD2/SCL20	P43/RxD2/INTPR2/ SDA20[Note 2]
0	0	1	0	1	1	0	0	×[Note 3]	×[Note 3]	×[Note 3]	×[Note 3]	Operation stop mode	P42	P43/RxD2/INTPR2
	1	0												P43
1	0	1	1	0/1[Note 4]	1	1	0	0	1	×[Note 3]	×[Note 3]	UART2 transmission[Note 5]	TxD2	P43/RxD2/INTPR2
0	1	0	0	0/1[Note 6]	0/1[Note 6]	0	0	0	1	0	1	IIC20 start condition	SCL20	SDA20
						1	0							
						0	1							
1			1	0/1[Note 4]	0/1[Note 4]	1	0	0	1	0	1	IIC20 address field transmission	SCL20	SDA20
			1	0/1[Note 4]	0/1[Note 4]	1	0	0	1	0	1	IIC20 data transmission	SCL20	SDA20
			1	0/1[Note 3]	0/1[Note 3]	0	1	0	1	0	1	IIC20 data reception	SCL20	SDA20
0			0	0/1[Note 7]	0/1[Note 7]	0	0	0	1	1	×	IIC20 stop condition	SCL20	SDA20
						1	0							
						0	1							

Notes 1. The SE2 register is a read-only status register which is set using the SS2 and ST2 registers.

2. When channel 1 of unit 2 is set to UART2 reception, this pin becomes an RxD2 function pin (see **Table 11-14**). In this case, operation stop mode or UART2 transmission must be selected for channel 2 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, see **11.3 (12) Serial output register m (SOm)**.

5. When using UART2 transmission and reception in a pair, set channel 1 of unit 2 to UART2 reception (see **Table 11-14**).

6. Set the CKO20 bit to 1 before a start condition is generated. Clear the SO20 bit from 1 to 0 when the start condition is generated.

7. Set the CKO20 bit to 1 before a stop condition is generated. Clear the SO20 bit from 0 to 1 when the stop condition is generated.

Remark X: Don’t care

Table 11-14. Relationship Between Register Settings and Pins (Channel 1 of Unit 2: UART2 Reception)

SE2_1 [Note 1]	MD212	MD211	SOE2_1	SO21	CKO21	TXE21	RXE21	PM4_3 [Note 2]	P4_3 [Note 2]	Operation Mode	Pin Function
											P43/RxD2/INTPR2/SDA20[Note 2]
0	0	1	0	1	1	0	0	× [Note 3]	× [Note 3]	Operation stop mode	P43
1	0	1	0	1	1	0	1	1	1	UART2 reception[Notes 4, 5]	RxD2/INTPR2

Notes 1. The SE2 register is a read-only status register which is set using the SS2 and ST2 registers.

2. When channel 1 of unit 2 is set to UART2 reception, this pin becomes an RxD2 function pin. In this case, set channel 0 of unit 2 to operation stop mode or UART2 transmission (see **Table 11-13**).

When channel 0 of unit 2 is set to IIC20, this pin cannot be used as an RxD2 function pin. In this case, set channel 1 of unit 2 to operation stop mode or IIC11.

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 2 to UART2 transmission (see **Table 11-13**).

5. The SMR20 register of channel 0 of unit 2 must also be set during UART2 reception. For details, see **11.6.2 (1) Register setting**.

Remark X: Don’t care

CHAPTER 12 ASYNCHRONOUS SERIAL INTERFACE LIN-UART (UARTF)

In the 78K0R/Hx3, an asynchronous serial interface LIN-UART (UARTF) is provided.

12.1 Features

○ Maximum transfer rate: 1 Mbps (using dedicated baud rate generator)

○ Full-duplex communication:	Internal LIN-UART receive data register n (UFnRX)
	Internal LIN-UART transmit data register n (UFnTX)
○ 2-pin configuration:	LTxDn: Transmit data output pin
	LRxDn: Receive data input pin

○ Error detection function of receiving data

- Parity error
- Framing error
- Overrun error
- Function to detect consistency errors in LIN communication data
- Function to detect successful BF reception
- ID parity error
- Checksum error
- Response preparation error
- ID match function
- Expansion bit detection function

○ Interrupt sources: 3

- Reception complete interrupt (INTLRn)
- Transmission interrupt (INTLTn)
- Status interrupt (INTLSn)

○ Character length: 7, 8 bits

○ Communication with 9-bit data length possible by expansion bit setting

○ When an expansion bit is at the expected level, the received data can be compared with 8-bit data set in a register in advance

○ Internal 3-bit prescaler

○ Parity function: Odd, even, 0, none

○ Transmission stop bit: 1, 2 bits

○ On-chip dedicated baud rate generator

○ MSB-/LSB-first transfer selectable

○ Transmit/receive data inverted input/output possible

○ Guarantee for stop bit of reception (suspension of transmission start during stop bit of reception when starting transmission possible)

Remark n = 0, 1

○ Transmission/reception function in a LIN (Local Interconnect Network) communication format

- 13 to 20 bits selectable for BF transmission
- Recognition of 11 bits or more in the LIN communication format possible for BF reception
- BF reception flag provided
- Detection of new BF reception possible during data communication
- Function to check consistency of transmit data provided (function to detect mismatches by comparing transmit data and receive data)
- Automatic slave baud rate setting
- Automatic checksum generation function provided (function to automatically calculate the checksum during response transmission or response reception)
- ID parity check function provided (function to automatically check the parity bit of the PID received)

Remark LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

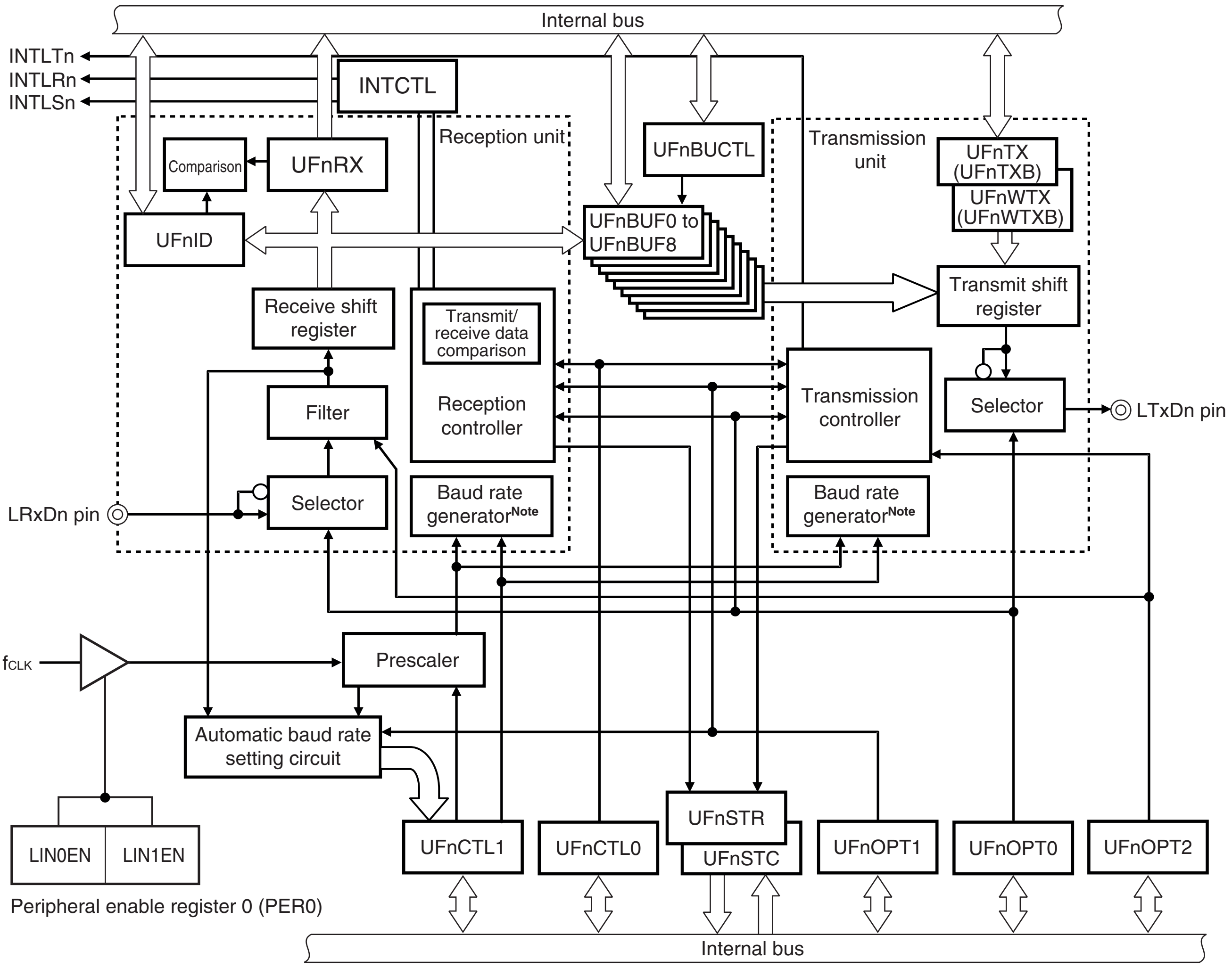
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±14% or less.

12.2 Configuration

<R> **Figure 12-1. Block Diagram of Asynchronous Serial Interface LIN-UART**



Note For the configuration of the baud rate generator, see **Figure 12-73 Configuration of Baud Rate Generator**.

Remark n = 0, 1

LIN-UART consists of the following hardware units.

Table 12-1. Configuration of LIN-UARTn

Item	Configuration
Registers	Peripheral enable register 0 (PER0) LIN-UARTn control registers 0, 1 (UFnCTL0, UFnCTL1) LIN-UARTn option registers 0 to 2 (UFnOPT0 to UFnOPT2) LIN-UARTn status register (UFnSTR) LIN-UARTn status clear register (UFnSTC) LIN-UARTn receive shift register LIN-UARTn receive data register (UFnRX) LIN-UARTn 8-bit receive data register (UFnRXB) LIN-UARTn transmit shift register LIN-UARTn transmit data register (UFnTX) LIN-UARTn 8-bit transmit data register (UFnTXB) LIN-UARTn wait transmit data register (UFnWTX) LIN-UARTn 8-bit wait transmit data register (UFnWTXB) LIN-UARTn ID setting register (UFnID) LIN-UARTn buffer registers 0 to 8 (UFnBUF0 to UFnBUF8) LIN-UARTn buffer control register (UFnBUCTL) Serial communication pin select register (STSEL) Port mode registers 1, 7 (PM1, PM7)

Remark n = 0, 1

12.3 Control Registers

(1) Peripheral enable register 0 (PER0)

The PER0 register is used to set whether to use each peripheral hardware unit. Power consumption and noise can be reduced, because clock supply will be stopped for the hardware not to be used.

When using LIN-UART, be sure to set the bits of the LIN-UART to be used (bit 6 (LIN1EN) and bit 5 (LIN0EN)) to 1.

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

Reset signal generation sets this register to 00H.

Figure 12-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	ADCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN

LIN1EN	LIN-UART1 input clock control
0	Stops input clock supply. • Writing to SFR to be used with LIN-UART1 is disabled. • LIN-UART1 is in reset state.
1	Supplies input clock. • Reading from and writing to SFR to be used with LIN-UART1 is enabled.

LIN0EN	LIN-UART0 input clock control
0	Stops input clock supply. • Writing to SFR to be used with LIN-UART0 is disabled. • LIN-UART0 is in reset state.
1	Supplies input clock. • Reading from and writing to SFR to be used with LIN-UART0 is enabled.

Caution In the 78K0R/HC3, be sure to clear bit 2 to “0”.

(2) LIN-UARTn control register 0 (UFnCTL0)

The UFnCTL0 register is an 8-bit register that controls serial communication operation of LIN-UARTn.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 10H.

<R> **Figure 12-3. Format of LIN-UARTn Control Register 0 (UFnCTL0) (1/2)**

Address: F0240H (UF0CTL0), F0260H (UF1CTL0) After reset: 10H R/W

	7	<6>	<5>	4	3	2	1	0
UFnCTL0	0	UFnTXE	UFnRXE	UFnDIR	UFnPS1	UFnPS0	UFnCL	UFnSL
(n = 0, 1)								

UFnTXE	Transmission operation enable
0	Stops transmission operation.
1	Enables transmission operation.
The level of a LTxDn pin is not concerned with a setup of a UFnTXE bit, but a setup of the UFnTDL bit of UFnOPT0 register is reflected.	

UFnRXE	Reception operation enable
0	Stops reception operation.
1	Enables reception operation.
During reception operation, if 0 is set to a UF0RXE bit, reception operation will be aborted. The interrupt request signal of the completion of reception is not outputted, and a receiving data register is not updated, either. In the automatic baud rate mode (UFnMD1, UFnMD0 = 11B), set to UFnRXE=1 after a receiving pin becomes high-level. If UFnRXE=1 is used on a low level, BF detection will be begun from the moment.	

UFnDIR	Communication direction mode (MSB/LSB) selection
0	MSB first
1	LSB first
• Rewriting is possible only when UFnTXE = UFnRXE = 0. • To perform transmission and reception in the LIN communication format, set the UFnDIR bit to “1”.	

Figure 12-3. Format of LIN-UARTn Control Register 0 (UFnCTL0) (2/2)

UFnPS1	UFnPS0	Parity selection during transmission	Parity selection during reception
0	0	No parity output	Reception with no parity
0	1	0 parity output	No parity check
1	0	Odd parity output	Odd parity check
1	1	Even parity output	Even parity check

- Rewriting is possible only when UFnTXE = UFnRXE = 0.
- If “Reception with no parity” or “Reception with 0 parity” is selected during reception, a parity check is not performed.
  Consequently, a status interrupt (INTLSn) is not generated with parity error, because the UFnPE bit of the UFnSTR register is not set.
- To perform transmission and reception in the LIN communication format, set the UFnPS1 and UFnPS0 bits to “00”.

UFnCL	Specification of data character length of 1 frame of transmit/receive data
0	7 bits
1	8 bits

- Rewriting is possible only when UFnTXE = UFnRXE = 0.
- To perform transmission and reception in the LIN communication format, set the UFnCL bit to “1”.

UFnSL	Specification of length of stop bit for transmit data
0	1 bit
1	2 bits

Rewriting is possible only when UFnTXE = UFnRXE = 0.

Caution During receive data framing error detection, only the first bit of the stop bits is checked, regardless of the value of the stop bit length select bit (UFnSL).

Remark For details of parity, see **12.5.7 Parity types and operations**.

(3) LIN-UARTn control register 1 (UFnCTL1)

See **12.10 (2) LIN-UARTn control register 1 (UFnCTL1)** for details.

(4) LIN-UARTn option register 0 (UFnOPT0)

The UFnOPT0 register is an 8-bit register that controls serial communication operation of LIN-UARTn.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 14H.

Figure 12-4. Format of LIN-UARTn Option Register 0 (UFnOPT0) (1/3)

Address: F0241H (UF0OPT0), F0261H (UF1OPT0) After reset: 14H R/W

	<7>	<6>	<5>	4	3	2	1	0
UFnOPT0 (n = 0, 1)	UFnBRF	UFnBRT	UFnBTT	UFnBLS2	UFnBLS1	UFnBLS0	UFnTDL	UFnRDL

UFnBRF	BF reception flag
0	When the UFnCTL0.UFnRXE = 0 is set. Also upon normal end of BF reception.
1	While waiting for successful BF reception (when the UFnBRT bit is set)

- BF (Break Field) reception is judged during LIN communication.
- The UFnBRF bit retains “1” when a BF reception error occurs, and is cleared to “0” when BF reception is started again and ends normally. It cannot be cleared by instruction.
- The UFnBRF bit is read-only.

<R>

Caution When the UFnBRF bit is 1, whether BF reception has ended normally can be judged by checking whether the low-level period is at least 11 bits, when a high level, including noise, is input to the receive input data even for a moment. If the low-level period is at least 11 bits, BF reception is judged to be performed successfully.

During communication, in the case of the BF receivable mode (UFnMD1, UFnMD0 = 10B), if BF reception success flag (UFnBSF) is “1” when status interruption is detected, it can be checked that BF reception has been completed normally.

In addition, during communication, in the case of the BF receivable mode, even if it sets up BF reception trigger bit, the completion interrupt of reception does not occur, but if the UFnBRF flag is “0” at the time of the status interruption detection after a setup, it can check similarly.

In either case, operation is performed as a normal UART reception from the next reception after BF reception has been performed successfully.

UFnBRT	BF reception trigger
0	–
1	BF reception trigger

- This is the BF reception trigger bit during LIN communication, and when read, “0” is always read. For BF reception, set (1) the UFnBRT bit to enable BF reception.
- Set the UFnBRT bit after having set UFnCTL0.UFnRXE to “1”.
- The status flag will not be updated, an interrupt request signal will not be generated, and data will not be stored.
- This bit can only be set again when the UFnBRF bit is 0.
- When BF reception is enabled during communication, BF reception is detected as the low-level period between when the UFnBRT bit is set and when the rising edge of the reception input data is detected. Therefore, a BF will be detected even if the UFnBRT bit is set during BF reception.

Cautions 1. To release a BF reception enable state without receiving a BF, UFnRXE must be cleared to 0.

2. Transmitting data while UFnDCS and UFnBRF are “1” is prohibited. BF transmission, however, can be performed.

3. Setting the UFnBRT bit in automatic baud rate mode (UFnMD1, UFnMD0 = 11B) is prohibited.

Figure 12-4. Format of LIN-UARTn Option Register 0 (UFnOPT0) (2/3)

<R>

UFnBTT	BF transmission trigger
0	–
1	BF transmission trigger

- This is the BF transmission trigger bit during LIN communication, and when read, “0” is always read.
- Set the UFnBTT bit after having set UFnCTL0.UFnTXE to “1”.

Cautions 1. During data transmission, it is prohibition to perform simultaneously a setup of the data transmitted to the next and a setup of this bit.

Also, even if the UFnBTT bit is set during a BF transmission, it is invalid (a BF transmission is performed once and ends).

2. Completion of a BF transmission can be judged by checking that the UFnTSF bit is “0” after the BF transmission trigger bit has been set. If the next transmit data has been written to the UFnTX register during the BF transmission, however, the UFnTSF bit will not be cleared when transmitting the BF has been completed, but will retain “1”.

When in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B), completion of a BF transmission can also be judged by checking that the successful BF reception flag (UFnBSF) is “1” after a status interrupt has been detected.

3. Setting the UFnBTT bit is prohibited in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

UFnBLS2	UFnBLS1	UFnBLS0	BF length selection bit
1	0	1	13-bit output (reset value)
1	1	0	14-bit output
1	1	1	15-bit output
0	0	0	16-bit output
0	0	1	17-bit output
0	1	0	18-bit output
0	1	1	19-bit output
1	0	0	20-bit output

This bit can be set when UFnCTL0.UFnTXE is “0”.

UFnTDL	Transmit data level bit
0	Normal output of transfer data
1	Inverted output of transfer data

- The LTxDn output value can be inverted by using the UFnTDL bit.
- This bit can be set when UFnCTL0.UFnTXE is “0”.

Cautions 1. The LTxDn output level is inverted by controlling the UFnTDL bit, regardless of the value of the UFnTXE bit. Consequently, if the UFnTDL bit is set to “1” even when operation is disabled, the LTxDn output becomes low level.

2. To perform transmission and reception in the LIN communication format, set UFnTDL to “0”.

Figure 12-4. Format of LIN-UARTn Option Register 0 (UFnOPT0) (3/3)

UFnRDL	Receive data level bit
0	Normal input of transfer data
1	Inverted input of transfer data

- The LRxDn input value can be inverted by using the UFnRDL bit.
- This bit can be set when UFnCTL0.UFnRXE is “0”.

Cautions 1. Be sure to enable reception (UFnRXE = 1) after having changed the UFnRDL bit. When the UFnRDL bit is changed after reception has been enabled, the start bit will be falsely detected, depending on the pin level at that time.

2. To perform transmission and reception in the LIN communication format, set UFnRDL to “0”.

(5) LIN-UARTn option register 1 (UFnOPT1)

The UFnOPT1 register is an 8-bit register that controls serial communication operation of LIN-UARTn.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

Caution Set the UFnOPT1 register when UFnTXE and UFnRXE are "0". Only the UFnEBC bit, however, can be changed even if UFnTXE is "1" or UFnRXE is "1". See 12.8.3 Expansion bit mode reception (with data comparison) for details.

Figure 12-5. Format of LIN-UARTn Option Register 1 (UFnOPT1) (1/3)

Address: F0244H (UF0OPT1), F0264H (UF1OPT1) After reset: 00H R/W

	7	6	<5>	4	3	2	1	0
UFnOPT1	UFnEBE	UFnEBL	UFnEBC	UFnIPCS	UFnACE	UFnMD1	UFnMD0	UFnDCS
(n = 0, 1)								

UFnEBE	Expansion bit enable bit
0	Disables expansion bit operation. (Transmission and reception are performed in the data length (7, 8 bits) set to UFnCTL0.UFnCL.)
1	Enables expansion bit operation. (Transmission and reception are performed in data length (9 bits) when UFnCTL0.UFnCL is "1".)
Cautions	**1. To perform transmission and reception in 9-bit units by setting (1) the UFnEBE bit, the data length must be set to 8 bits (UFnCL = 1). If the data length is set to 7 bits (UFnCL = 0), the setting of the UFnEBE bit will be invalid.** **2. To perform transmission and reception in the LIN communication format, set UFnEBE to "0".**
<R> **Remark**	Expansion bit is the target of a parity check.

UFnEBL	Expansion bit detection level select bit
0	Selects expansion bit value "0" as expansion bit detection level.
1	Selects expansion bit value "1" as expansion bit detection level.
	If the level selected by the UFnEBL bit is detected as the expansion bit when the expansion bit has been enabled (UFnCL = UFnEBE = 1), a status interrupt request signal (INTLSn) will be generated and an expansion bit detection flag (UEnEBD) will be set. If the inversion level is detected as the expansion bit, a reception complete interrupt request signal (INTLRn) will be generated, but an expansion bit detection flag will not be set.
Remark	The UFnEBL bit becomes valid only if UFnCL = UFnEBE = 1. See **12.8.2 Expansion bit mode reception (no data comparison)** and **12.8.3 Expansion bit mode reception (with data comparison)** for details.

Figure 12-5. Format of LIN-UARTn Option Register 1 (UFnOPT1) (2/3)

UFnEBC	Expansion bit data comparison enable bit
0	No comparison (INTLRn or INTLSn is always generated upon completion of data reception.)
1	Compares UFnRX register and UFnID register when the level selected for the UFnEBL bit has been detected as the expansion bit. (INTLSn is generated only when the UFnRX register and UFnID register have matched.)

The UFnEBC bit is used to enable comparison between the received data and the UFnID register when the expansion bit has been enabled (UFnCL = UFnEBE = 1).

Remark The UFnEBC bit becomes valid only if UFnCL = UFnEBE = 1. See **12.8.2 Expansion bit mode reception (no data comparison)** and **12.8.3 Expansion bit mode reception (with data comparison)** for details.

UFnIPCS	ID parity check select bit
0	No automatic ID parity check (Calculating the parity of the PID by using software and checking are required.)
1	Automatic ID parity check

- The UFnIPCS bit is used to select how to handle automatic checking of the parity bit of the received PID, when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).
- If UFnIPCS is “1”, the parity bit is checked when the PID received in LIN communication is stored into the UFnID register. When an incorrect result has been detected, an ID parity error flag (UFnIPE) will be set and a status interrupt request signal (INTLSn) will be generated.

Remark The UFnIPCS bit becomes valid only in the automatic baud rate mode (UFnMD1, UFnMD0 = 11B). See **12.7.3 ID parity check function** for details.

UFnACE	Automatic checksum enable bit
0	Disables automatic checksum calculation. Response transmission: Checksum must be calculated by using software and set to a buffer. Response reception: Checksum must be calculated from the data stored into the buffer by using software, and compared and checked with the checksum obtained via communication.
1	Enables automatic checksum calculation. Response transmission: Checksum is automatically calculated from the data set to a buffer and is automatically appended at the end of response transmission. Response reception: Checksum is automatically calculated from the data stored into the buffer and is automatically compared and checked with the checksum obtained via communication.

- The UFnACE bit is used to select how to handle automatic checksum calculation during response transmission and response reception, when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).
- When response reception is performed while UFnACE is “1”, the checksum received in LIN communication will be checked when it is stored into a receive buffer. When an incorrect result has been detected, a checksum error flag (UFnCSE) will be set and a status interrupt request signal (INTLSn) will be generated.

Remark The UFnACE bit becomes valid only in the automatic baud rate mode (UFnMD1, UFnMD0 = 11B). See **12.7.4 Automatic checksum function** for details.

Figure 12-5. Format of LIN-UARTn Option Register 1 (UFnOPT1) (3/3)

UFnMD1	UFnMD0	LIN-UART operation mode select bit
0	0	Normal UART mode
0	1	Setting prohibited
1	0	LIN communication: BF reception enable mode during communication Detects a new Break Field during data communication. (When a low level has been detected at the stop bit position, a wait is performed until the next high level is detected and a new BF reception is recognized if the low-level period is at least 11 bits.)
1	1	LIN communication: Automatic baud rate mode

Cautions 1. Setting to automatic baud rate mode (UFnMD1, UFnMD0 = 11B) is prohibited for a LIN communication master.

2. Be sure to also set the UFnDCS bit to "1" when in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B) or in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

Remark When in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B) during LIN communication, set TMLINn to 1 and select the input signal of the serial data input pin (LRxDn) as a timer input.

UFnDCS	Data consistency check select bit
0	Does not check data consistency.
1	Checks data consistency.

- The UFnDCS bit is used to select how to handle a data consistency check when transmitting data via LIN communication. For details, see **12.5.8 Data consistency check**.
- When UFnDCS is "1", transmit data and receive data will be compared when transmitting data via LIN communication. When a mismatch is detected, a data consistency error flag (UFnDCE) will be set and a status interrupt request signal (INTLSn) will be generated.

Cautions 1. When using LIN communication, the UFnDCS bit can be set. Otherwise, clear the UFnDCS bit to "0".

2. When setting (1) the UFnDCS bit, fix the data bit length to 8 bits. Appending a parity bit is prohibited.

3. Be sure to also set the UFnDCS bit to "1" when in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B) or in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

(6) LIN-UARTn option register 2 (UFnOPT2)

The UFnOPT2 register is an 8-bit register that controls serial communication operation of LIN-UARTn.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

Figure 12-6. Format of LIN-UARTn Option Register 2 (UFnOPT2)

Address: F0245H (UF0OPT2), F0265H (UF1OPT2) After reset: 00H R/W

	7	6	5	4	3	2	1	<0>
UFnOPT2 (n = 0, 1)	0	0	0	0	0	0	UFnRXFL	UFnITS

UFnRXFL	Bit to select use of receive data noise filter
0	Uses noise filter.
1	Does not use noise filter.
The UFnRXFL bit is used to select use of the noise filter. See **12.9 Receive Data Noise Filter** for details. **Caution Be sure to set the UFnRXFL bit when UFnCTL0.UFnRXE is “0”.**	

UFnITS	Transmission interrupt (INTLTn) generation timing select bit
0	Outputs transmission interrupt request upon transmission start.
1	Outputs transmission interrupt request upon transmission completion.
Caution Be sure to set the UFnITS bit when UFnCTL0.UFnTXE is “0”. **The UFnITS bit can be changed to 1 after transmission of the last data is started only when completion of transmitting the last data must be known during successive transmission (UFnITS = 0). However, the change must be completed before the transmission is completed.**	

(7) LIN-UARTn status register (UFnSTR)

The UFnSTR register is a 16-bit register that displays the LIN-UARTn communication status and reception error contents.

This register is read-only, in 16-bit units.

Reset sets this register to 0000H.

Caution Flags other than the UFnTSF and UFnRSF flags are retained until the target bits of the LIN-UARTn status clear register (UFnSTC) are written ("1") and then cleared. To clear a status flag, use a 16-bit manipulation instruction to write ("1") and clear the target bits of the LIN-UARTn status clear register (UFnSTC).

Figure 12-7. Format of LIN-UARTn Status Register (UFnSTR) (1/6)

Address: F0246H, F0247H (UF0STR), F0266H, F0267H (UF1STR) After reset: 0000H R

	15	14	13	12	11	10	9	8
UFnSTR	0	UFnIPE	UFnCSE	UFnRPE	UFnHDC	UFnBUC	UFnIDM	UFnEBD
(n = 0, 1)	7	6	5	4	3	2	1	0
	UFnTSF	UFnRSF	0	UFnBSF	UFnDCE	UFnPE	UFnFE	UFnOVE

UFnIPE	ID parity error flag
0	No ID parity error has occurred.
1	An ID parity error has occurred. <ID parity error source> Parity of received PID is incorrect
• The UFnIPE bit is a flag indicating the check status by the ID parity check function. It becomes "1", if the parity of the received PID is incorrect when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B). See **12.7.3 ID parity check function** for details. • The UFnIPE bit will not be cleared until "1" is written to the UFnCLIPE bit of the UFnSTC register, because the UFnIPE bit is a cumulative flag. It will not be set if the ID parity check function has been disabled (UFnIPCS = 0).	

Figure 12-7. Format of LIN-UARTn Status Register (UFnSTR) (2/6)

UFnCSE	Checksum error flag
0	No checksum error has occurred.
1	A checksum error has occurred. <Checksum error source> Result of comparing checksum automatically calculated from data stored into buffer and checksum obtained via communication is incorrect during response reception

- The UFnCSE bit is a flag indicating the check status by the automatic checksum function. It becomes “1” if the received checksum is incorrect when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B) and during response reception. See **12.7.4 Automatic checksum function** for details.
- The UFnCSE bit will not be cleared until “1” is written to the UFnCLCSE bit of the UFnSTC register, because the UFnCSE bit is a cumulative flag. It will not be set if the automatic checksum function has been disabled (UFnACE = 0).

Cautions 1. The check sum error flag will not be set during response transmission. Perform a data consistency check to check for errors.

2. Receive data will be stored in the UFnRX register during response transmission. However, no overrun error will be set, even if the receive data is not read. Consequently, the received check sum can be checked by reading the UFnRX register after the reception completion interrupt has occurred.

UFnRPE	Response preparation error flag
0	No response preparation error has occurred.
1	A response preparation error has occurred. < Response preparation error source> Response could not be prepared before completion of receiving first byte of receive data after header reception

- The UFnRPE bit is a flag indicating the check status by the response preparation detection function. It becomes “1”, if a response (setting of UFnNO, UFnRRQ bits) could not be prepared in automatic baud rate mode (UFnMD1, UFnMD0 = 11B). See **12.7.2 Response preparation error detection function** for details.
- The UFnRPE bit will not be cleared until “1” is written to the UFnCLRPE bit of the UFnSTC register, because the UFnRPE bit is a cumulative flag. It will not be set when not in automatic baud rate mode (UFnMD1, UFnMD0 = 00B or 10B).

UFnHDC	Header reception completion flag
0	Header reception is not completed.
1	Receiving header has been completed

- The UFnHDC bit is a flag indicating completion of receiving a header. It becomes “1” when receiving the header has been completed when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B). See **12.7.1 Automatic baud rate setting function** for details.
- The UFnHDC bit will not be cleared until “1” is written to the UFnCLHDC bit of the UFnSTC register, because the UFnHDC bit is a cumulative flag. It will not be set when not in automatic baud rate mode (UFnMD1, UFnMD0 = 00B or 10B).

Caution This flag will not be set by an error during PID reception.

Figure 12-7. Format of LIN-UARTn Status Register (UFnSTR) (3/6)

UFnBUC	Buffer transmission/reception completion flag
0	Buffer transmission/reception is not completed.
1	Buffer transmission/reception is completed <Buffer transmission/reception completion condition> The set number of data is transmitted or received. (only when transmitted when in normal UART mode)

- The UFnBUC bit is a flag indicating the data transmission and reception status of a buffer. It becomes “1” when the set number of data items have been transmitted or received without an error occurring. See **12.6.1 UART buffer mode transmission** and **12.7 LIN Communication Automatic Baud Rate Mode** for details.
- The UFnBUC bit will not be cleared until “1” is written to the UFnCLBUC bit of the UFnSTC register, because the UFnBUC bit is a cumulative flag. It will be set only when in normal UART mode (UFnMD1, UFnMD0 = 00B) or automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

UFnIDM	ID match flag
0	The ID does not match.
1	The IDdoes match <ID match condition> When 8 bits of receive data, excluding expansion bit, have matched with UFnID register value set in advance

- The UFnIDM bit is a flag indicating the result of comparing the 8 bits of receive data, excluding the expansion bit, and the UFnID register value set in advance when expansion bit data comparison has been enabled (UFnEBC = 1) by enabling the expansion bit (UFnCL = UFnEBE = 1). The comparison will be performed with the data for which the level set by using the expansion bit detection level select bit (UFnEBL) has been detected. The UFnIDM bit becomes “1” when the comparison result has matched. See **12.8.3 Expansion bit mode reception (with data comparison)** for details.
- The UFnIDM bit will not be cleared until “1” is written to the UFnCLIDM bit of the UFnSTC register, because the UFnIDM bit is a cumulative flag. It will not be set when the expansion bit has not been enabled and expansion bit data comparison has not been enabled (UFnCL = UFnEBE = UFnEBC = 1).

UFnEBD	Expansion bit detection flag
0	An extension bit is not detected
1	An extension bit is detected <Expansion bit detection condition> When level set by using expansion bit detection level select bit (UFnEBL) has been detected for expansion bit

- The UFnEBD bit is a flag indicating detection of the level set by using the expansion bit detection level select bit (UFnEBL) when the expansion bit has been enabled (UFnCL = UFnEBE = 1). It becomes “1” when the setting level has been detected. See **12.8.2 Expansion bit mode reception (no data comparison)** and **12.8.3 Expansion bit mode reception (with data comparison)** for details.
- The UFnEBD bit will not be cleared until “1” is written to the UFnCLEBD bit of the UFnSTC register, because the UFnEBD bit is a cumulative flag. It will not be set when the expansion bit has been disabled (UFnEBE = 0).

Figure 12-7. Format of LIN-UARTn Status Register (UFnSTR) (4/6)

<table>
<tr><th>UFnTSF</th><th>Transmission status flag</th></tr>
<tr><td>0</td><td>A transmit operation is not performed.
<Transmission stop condition>
• When UFnCTL0.UFnTXE has been cleared to “0”
&lt;R&gt; • When there is no data transmitted to the next in each following register and BF transmission is not set up after the completion of transmission. (UFnBTT is not set) UFnTX, UFnWTX, UFnBUF0 to UFnBUF8
• When there was no next transmit data in UFnTX, UFnWTX, UFnBUF0 to UFnBUF8 bit after BF transmission has ended
• When the transmission after data consistency error detection is completed</td></tr>
<tr><td>1</td><td>A transmit operation is performed.
<Transmission start condition>
• Writes to UFnTX, UFnWTX register
• When BF transmit trigger bit (UFnBTT) has been set[Note]
• When the transmission request bit (UFnTRQ) is set</td></tr>
<tr><td colspan="2">• The UFnTSF bit is always “1” when successive transmission is performed.
• To initialize the transmission unit, check that UFnTSF is “0” before performing initialization. If initialization is performed while UFnTSF = 1, the transmission will be aborted midway.
• If a BF is detected in BF reception enabled mode during communication and when transmitting data, or if a BF/SF is detected in automatic baud rate mode and when transmitting data, the UFnDCE flag will be set and the UFnTSF bit will be cleared when a status interrupt (INTLSn) is issued.

<b>Note</b> Only during BF period</td></tr>
</table>

<table>
<tr><th>UFnRSF</th><th>Reception status flag</th></tr>
<tr><td>0</td><td>A receive operation is not performed.
<Reception stop condition>
• When UFnCTL0.UFnRXE has been cleared to “0”
• When at sampling point of stop bit (first bit) during reception
• When UFnBRT = 1 is set
• When a BF is detected in BF reception enabled mode during communication
• When a BF/SF is detected in automatic baud rate mode</td></tr>
<tr><td>1</td><td>A receive operation is performed.
<Reception start condition>
When a start bit is detected (when it is detected that the data is 0 at the sampling point of the bit after the LRxDn falling edge is detected)</td></tr>
<tr><td colspan="2">To initialize the reception unit, check that UFnRSF is “0” before performing initialization. If initialization is performed while UFnRSF = 1, the rception will be aborted midway.</td></tr>
</table>

Figure 12-7. Format of LIN-UARTn Status Register (UFnSTR) (5/6)

UFnBSF	Successful BF reception flag
0	BF reception is not successfully performed.
1	BF reception is successfully performed. <BF reception stop condition> When successive low levels (BF) of at least 11 bits have been received

- The UFnBSF bit is a flag indicating that receiving a BF has been performed successfully. It becomes "1" when successive low levels (BF) of at least 11 bits have been received when in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B) (This occurs at the same time as the status interrupt (INTLSn) is issued upon the detection of the rising edge of the LRXDn pin.).
- The start of a new frame slot must be checked by reading the UFnBSF bit via status interrupt servicing, because the BF may also be received during data communication when in BF reception enable mode during communication.
- The UFnBSF bit will not be cleared until "1" is written to the UFnCLBSF bit of the UFnSTC register, because the UFnBSF bit is a cumulative flag. It will not be set when not in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B).

UFnDCE	Data consistency error flag
0	No data consistency error has occurred.
1	A data consistency error has occurred. <Data consistency error source> When transmit data and receive data do not match in LIN communication

- When the data consistency check select bit is set (UFnDCS = 1), the transmit data and receive data are compared upon data transmission. The UFnDCE bit becomes "1" at the same time as the status interrupt (INTLSn) is issued when a mismatch has been detected.
- The UFnDCE bit will not be cleared until "1" is written to the UFnCLDCE bit of the UFnSTC register, because the UFnDCE bit is a cumulative flag. When UFnDCS is "0", the UFnDCE bit will not be set.

Caution The next transfer will not be performed if a data consistency error is detected. See 12.5.8 Data consistency check for details.

UFnPE	Parity error flag
0	No parity error has occurred.
1	A parity error has occurred. <Parity error source> When parity of data and parity bit do not match during reception

- The operation of the UFnPE bit determines on the settings of the UFnPS1 and UFnPS0 bits.
- <R> The UFnPE bit will not be cleared until "1" is written to the UFnCLPE bit of the UFnSTC register or "0" is written to the UFnRXE bit of the UFnCTL0 register, because the UFnPE bit is a cumulative flag. When UFnPS1 and UFnPS0 are "0xB", the UFnPE bit will not be set. (x: Don't care)

Figure 12-7. Format of LIN-UARTn Status Register (UFnSTR) (6/6)

UFnFE	Framing error flag
0	No framing error has occurred.
1	A framing error has occurred. < Framing error source> When no stop bit is detected during reception

- Only the first bit of the receive data stop bits is checked, regardless of the setting value of the UFnSL bit.
- The UFnFE bit will not be cleared until “1” is written to the UFnCLFE bit of the UFnSTC register or “0” is written to the UFnRXE bit of the UFnCTL0 register, because the UFnFE bit is a cumulative flag.

UFnOVE	Overrun error flag
0	No overrun error has occurred.
1	An overrun error has occurred. < Overrun error source> When receive data has been stored into the UFnRX register and the next receive operation is completed before that receive data has been read

- When an overrun error has occurred, the data is discarded without the next receive data being written to the UFnRX register.
- The UFnFE bit will not be cleared until “1” is written to the UFnCLFE bit of the UFnSTC register or “0” is written to the UFnRXE bit of the UFnCTL0 register, because the UFnFE bit is a cumulative flag. It will not be set in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

Caution If no status interrupt due to an ID mismatch is issued while expansion bit data comparison is enabled (UFnEBE = 1 and UFnEBC = 1), as receive data will not be stored in the UFnRX register, the UFnOVE flag will not be set even if the receive data is not read. Furthermore, when transmitting in automatic baud rate mode, the receive data will be always stored in the UFnRX register, but the UFnOVE flag will not be set even if the receive data is not read.

(8) LIN-UARTn status clear register (UFnSTC)

The UFnSTC register is a 16-bit register that is used to clear an LIN-UARTn status flag.

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

Reset sets this register to 0000H.

Caution An LIN-UART status register (UFnSTR) flag can be cleared by writing “1” to a corresponding bit. 0 will be read if the bit is read.

Figure 12-8. Format of LIN-UARTn Status Clear Register (UFnSTC) (1/2)

Address: F0248H, F0249H (UF0STC), F0268H, F0269H (UF1STC) After reset: 0000H R/W

	15	14	13	12	11	10	9	8
UFnSTC	0	UFnCLIPE	UFnCLCSE	UFnCLRPE	UFnCLHDC	UFnCLBUC	UFnCLIDM	UFnCLEBD
(n = 0, 1)	7	6	5	4	3	2	1	0
	0	0	0	UFnCLBSF	UFnCLDCE	UFnCLPE	UFnCLFE	UFnCLOVE

UFnCLIPE	Channel n ID parity error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UFnIPE bit of the UFnSTR register.

UFnCLCSE	Channel n checksum error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UFnCSE bit of the UFnSTR register.

UFnCLRPE	Channel n response preparation error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UFnRPE bit of the UFnSTR register.

UFnCLHDC	Channel n header reception completion flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UFnHDC bit of the UFnSTR register.

UFnCLBUC	Channel n buffer transmission/reception completion flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UFnBUC bit of the UFnSTR register.

UFnCLIDM	Channel n ID match flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UFnIDM bit of the UFnSTR register.

UFnCLEBD	Channel n expansion bit detection flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UFnEBD bit of the UFnSTR register.

Figure 12-8. Format of LIN-UARTn Status Clear Register (UFnSTC) (2/2)

UFnCLBSF	Channel n successful BF reception flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UFnBSF bit of the UFnSTR register.

UFnCLDCE	Channel n data consistency error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UFnDCE bit of the UFnSTR register.

UFnCLPE	Channel n parity error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UFnPE bit of the UFnSTR register.

UFnCLFE	Channel n framing error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UFnFE bit of the UFnSTR register.

UFnCLOVE	Channel n overrun error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UFnOVE bit of the UFnSTR register.

(9) LIN-UARTn transmit data register (UFnTX)

The UFnTX register is a 16-bit register that is used to set transmit data.

This register can be read or written in 16-bit units. When the UFnTX register is read or written in 8-bit units, it can be accessed as the UFnTXB register.

When no buffer is used and no data consistency error has been detected (UFnDCE = 0) in a transmission enable state (UFnTXE = 1), transmission is started by writing transmit data to the UFnTX register.

When UFnEBE = 0, transmit data of a character length specified by the UFnCL bit will be transmitted.

When UFnEBE = UFnCL = 1, transmit data of 9-bit length will be transmitted. See **12.5.1 Data format** for the transmit data format.

The last data written to the UFnTX register before it is loaded to the transmit shift register is to be transmitted. When UFnITS is “0”, successive transmission can be performed by writing the next transmit data to the UFnTX register after a transmission interrupt request has been generated. When the next transmit data is written before a transmission interrupt request is generated, the previously written data will be overwritten and only the subsequent data will be transmitted.

Reset input sets this register to 0000H.

Figure 12-9. Format of LIN-UARTn Transmit Data Register (UFnTX)

Address: FFF48H, FFF49H (UF0TX), FFF4CH, FFF4DH (UF1TX) After reset: 0000H R/W

	15	14	13	12	11	10	9	8
UFnTX	0	0	0	0	0	0	0	UFnTX8
(n = 0, 1)	7	6	5	4	3	2	1	0
	UFnTX7	UFnTX6	UFnTX5	UFnTX4	UFnTX3	UFnTX2	UFnTX1	UFnTX0

When the data length is specified as 7 bits (UFnCL = 0):

- During LSB-first transmission, bits 6 to 0 of the UFnTX register will be transferred as transmit data.
- During MSB-first transmission, bits 7 to 1 of the UFnTX register will be transferred as transmit data.

Cautions 1. If the UFnTX register is written while transmission is disabled (UFnTXE = 0), it will not operate as a transmission start trigger. Consequently, no transmission will be started, even if transmission is enabled after having written to the UFnTX register while transmission was disabled.

2. When the UFnTX register is written in 8-bit units (when the UFnTXB register is written), “0” is written to the UFnTX8 bit.

3. Writing to the UFnTX register is prohibited when using the UFnBUF0 to UFnBUF8 registers.

<R> **4. When using an auto check sum function, it is necessary to set 0000H to a UFnTX register.**

<R> **Remark** The UFnTX8 bit is an expansion bit when expansion bits are enabled (UFnEBE = UFnCL = 1).

(10) LIN-UARTn 8-bit transmit data register (UFnTXB)

The UFnTXB register is an 8-bit register that is used to set transmit data.

This register can be read or written in 8-bit units.

When no buffer is used and no data consistency error has been detected (UFnDCE = 0) in a transmission enable state (UFnTXE = 1), transmission is started by writing transmit data to the UFnTXB register.

When UFnEBE = 0, transmit data of a character length specified by the UFnCL bit will be transmitted.

The last data written to the UFnTXB register before it is loaded to the transmit shift register is to be transmitted. When UFnITS is "0", successive transmission can be performed by writing the next transmit data to the UFnTXB register after a transmission interrupt request has been generated. When the next transmit data is written before a transmission interrupt request is generated, the previously written data will be overwritten and only the subsequent data will be transmitted.

Reset input sets this register to 00H.

<R> **Figure 12-10. Format of LIN-UARTn 8-bit Transmit Data Register (UFnTXB)**

Address: FFF48H (UF0TXB), FFF4CH (UF1TXB) After reset: 00H R/W

	7	6	5	4	3	2	1	0
UFnTXB	UFnTX7	UFnTX6	UFnTX5	UFnTX4	UFnTX3	UFnTX2	UFnTX1	UFnTX0
(n = 0, 1)								

When the data length is specified as 7 bits (UFnCL = 0):

- During LSB-first transmission, bits 6 to 0 of the UFnTXB register will be transferred as transmit data.
- During MSB-first transmission, bits 7 to 1 of the UFnTXB register will be transferred as transmit data.

Cautions 1. If the UFnTXB register is written while transmission is disabled (UFnTXE = 0), it will not operate as a transmission start trigger. Consequently, no transmission will be started, even if transmission is enabled after having written to the UFnTXB register while transmission was disabled.

2. When the UFnTXB register is written, "0" is written to the UFnTX8 bit of UFnTX register.

3. Writing to the UFnTXB register is prohibited when using the UFnBUF0 to UFnBUF8 registers.

4. When using an auto check sum function, it is necessary to set 00H to a UFnTXB register.

(11) 8-bit transmit data register for LIN-UARTn wait (UFnWTX)

The UFnWTX register is a 16-bit register dedicated to delaying starting transmission until the stop bit of reception is completed during a LIN communication.

This register is write-only, in 16-bit units. When the UFnWTX register is write in 8-bit units, it can be accessed as the UFnWTXB register.

The stop bit length of reception when reception is switched to transmission is guaranteed for the UFnWTX register. See **12.5.11 Transmission start wait function** for details.

The UFnTX register value will be read when the UFnWTX register has been read.

Reset input sets this register to 0000H.

Figure 12-11. Format of 8-bit transmit data register for LIN-UARTn wait (UFnWTX)

Address: F024AH, F024BH (UF0WTX), F026AH, F026BH (UF1WTX) After reset: 0000H W

	15	14	13	12	11	10	9	8
UFnWTX	0	0	0	0	0	0	0	UFnWTX8
(n = 0, 1)	7	6	5	4	3	2	1	0
	UFnWTX7	UFnWTX6	UFnWTX5	UFnWTX4	UFnWTX3	UFnWTX2	UFnWTX1	UFnWTX0

Cautions 1. Writing to the UFnWTX register is prohibited other than when reception is switched to transmission (such as during transmission).

2. When the UFnWTX register is accessed in 8-bit units (when the UFnWTXB register is accessed), "0" is written to the UFnWTX8 bit.

<R> **3. Writing to the UFnWTX register is prohibited when using the UFnBUF0 to UFnBUF8 registers.**

Remark The UFnWTX8 bit is an expansion bit when expansion bits are enabled (UFnEBE = UFnCL = 1).

(12) LIN-UARTn 8-bit wait transmit data register (UFnWTXB)

The UFnWTXB register is an 8-bit register dedicated to delaying starting transmission until the stop bit of reception is completed during a LIN communication.

This register is write-only, in 8-bit units.

The stop bit length of reception when reception is switched to transmission is guaranteed for the UFnWTXB register.

See **12.5.11 Transmission start wait function** for details.

<R> The UFnTXB register value will be read when the UFnWTXB register has been read.

Reset input sets this register to 00H.

Figure 12-12. Format of LIN-UARTn 8-bit Wait Transmit Data Register (UFnWTXB)

Address: F024AH (UF0WTXB), F026AH (UF1WTXB) After reset: 00H W

	7	6	5	4	3	2	1	0
UFnWTXB	UFnWTX7	UFnWTX6	UFnWTX5	UFnWTX4	UFnWTX3	UFnWTX2	UFnWTX1	UFnWTX0

(n = 0, 1)

Cautions 1. Writing to the UFnWTXB register is prohibited other than when reception is switched to transmission (such as during transmission).

2. When the UFnWTXB register is accessed in 8-bit units (when the UFnWTXB register is accessed), "0" is written to the UFnWTX8 bit of UFnWTX register.

<R> **3. Writing to the UFnWTXB register is prohibited when using the UFnBUF0 to UFnBUF8 registers.**

(13) LIN-UARTn receive data register (UFnRX)

The UFnRX register is a 16-bit register that is used to store receive data.

Receive data of a character length specified by the UFnCL bit after reception completion will be stored into the UFnRX register when not in automatic baud rate mode (UFnMD1, UFnMD0 = 00B/10B) and when UFnEBE is “0”. When UFnEBE = UFnCL = 1, receive data of 9-bit length will be stored.

This register is read-only, in 16-bit units. When the UFnRX register is read in 8-bit units, it can be accessed as the UFnRX register.

Reset input sets this register to 0000H.

<R> **Figure 12-13. Format of LIN-UARTn Receive Data Register (UFnRX)**

Address: FFF4AH, FFF4BH (UF0RX), FFF4EH, FFF4CH (UF1RX) After reset: 0000H R

	15	14	13	12	11	10	9	8
UFnRX	0	0	0	0	0	0	0	UFnRX8
(n = 0, 1)	7	6	5	4	3	2	1	0
	UFnRX7	UFnRX6	UFnRX5	UFnRX4	UFnRX3	UFnRX2	UFnRX1	UFnRX0

When the data length is specified as 7 bits (UFnCL bit = 0):

- During LSB-first reception, receive data is transferred to bits 6 to 0 of the UFnRX register and the MSB always becomes “0”.
- During MSB-first reception, receive data is transferred to bits 7 to 1 of the UFnRX register and the LSB always becomes “0”.
- When an overrun error (UFnOVE = 1) has occurred, the receive data at that time will not be transferred to the UFnRX register.

Remark The UFnRX8 bit is an expansion bit when expansion bits are enabled (UFnEBE = UFnCL = 1).

(14) LIN-UARTn 8-bit receive data register (UFnRXB)

The UFnRXB register is an 8-bit register that is used to store receive data.

Receive data of a character length specified by the UFnCL bit after reception completion will be stored into the UFnRX register when not in automatic baud rate mode (UFnMD1, UFnMD0 = 00B/10B) and when UFnEBE is "0".

This register is read-only, in 8-bit units.

Reset input sets this register to 00H.

Figure 12-14. Format of LIN-UARTn 8-bit Receive Data Register (UFnRXB)

Address: FFF4AH (UF0RXB), FFF4EH (UF1RXB) After reset: 00H R

	7	6	5	4	3	2	1	0
UFnRXB	UFnRX7	UFnRX6	UFnRX5	UFnRX4	UFnRX3	UFnRX2	UFnRX1	UFnRX0

(n = 0, 1)

When the data length is specified as 7 bits (UFnCL bit = 0):

- During LSB-first reception, receive data is transferred to bits 6 to 0 of the UFnRX register and the MSB always becomes "0".
- During MSB-first reception, receive data is transferred to bits 7 to 1 of the UFnRX register and the LSB always becomes "0".
- When an overrun error (UFnOVE = 1) has occurred, the receive data at that time will not be transferred to the UFnRX register.

(15) LIN-UARTn ID setting register (UFnID)

The UFnID register is an 8-bit register that stored a PID that has been received when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B) and during a LIN communication. See **12.7 LIN Communication Automatic Baud Rate Mode** for details.

Also, when in normal UART mode (UFnMD1, UFnMD0 = 00B) and expansion bit data comparison is enabled (UFnCL = UFnEBE = UFnEBC = 1), the 8 bits (UFnRX7 to UFnRX0) of the receive data and the UFnID register are compared upon a match between the received expansion bit and the expansion bit detection level (UFnEBL). See **12.8.3 Expansion bit mode reception (with data comparison)** for details.

Be sure to execute LIN communication by setting the reception enable bit (the UFnRXE bit of the UFnCTL0 register) to 0 when specifying a comparison value, and then setting the bit to 1.

This register can be read or written in 8-bit units.

Reset input sets this register to 00H.

Figure 12-15. Format of LIN-UARTn ID Setting Register (UFnID)

Address: F024EH (UF0ID), F026EH (UF1ID) After reset: 00H R/W

	7	6	5	4	3	2	1	0
UFnID	UFnID7	UFnID6	UFnID5	UFnID4	UFnID3	UFnID2	UFnID1	UFnID0

(n = 0, 1)

<R> **Caution Set 00H to a UFnID register before communication when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B). The writing to a UFnID register is prohibition during communication in automatic baud rate mode.**

(16) LIN-UARTn buffer registers 0 to 8 (UFnBUF0 to UFnBUF8)

The UFnBUF0 to UFnBUF8 registers are 8-bit buffer registers.

These registers can be used when transmitting data in normal UART mode (UFnMD1 and UFnMD0 = 00B) and when transmitting and receiving data in automatic baud rate mode (UFnMD1 and UFnMD0 = 11B).

When in normal UART mode (UFnMD1, UFnMD0 = 00B), data will be sequentially transmitted from the UFnBUF0 register by setting the UFnTRQ bit.

When in automatic baud rate mode (UFnMD1, UFnMD0 = 11B) and during response transmission (UFnTRQ = 1), the transmit data in UFnBUF0 will be transmitted sequentially, but the received data will not be stored.

When in automatic baud rate mode (UFnMD1, UFnMD0 = 11B) and during response reception (UFnRRQ = 1), the received data will be stored sequentially, starting from the UFnBUF0 register.

See **12.6.1 UART buffer mode transmission** and **12.7 LIN Communication Automatic Baud Rate Mode** for details.

These registers can be read or written in 8-bit units.

Reset input sets these registers to 00H.

Figure 12-16. Format of LIN-UARTn Buffer Registers 0 to 8 (UFnBUF0 to UFnBUF8)

Address: F024FH (UF0BUF0), F0250H (UF0BUF1), After reset: 00H R/W
F0251H (UF0BUF2), F0252H (UF0BUF3),
F0253H (UF0BUF4), F0254H (UF0BUF5),
F0255H (UF0BUF5), F0256H (UF0BUF7),
F0257H (UF0BUF8)

F026FH (UF1BUF0), F0270H (UF1BUF1),
F0271H (UF1BUF2), F0272H (UF1BUF3),
F0273H (UF1BUF4), F0274H (UF1BUF5),
F0275H (UF1BUF6), F0276H (UF1BUF7),
F0277H (UF1BUF8)

	7	6	5	4	3	2	1	0
UFnBUFm								

(n = 0, 1, m = 0 to 8)

Caution These registers cannot be used when expansion bits are enabled (UFnEBE = UFnCL = 1).

(17) LIN-UARTn buffer control register (UFnBUCTL)

The UFnBUCTL register is a 16-bit register that controls a buffer.

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

See **12.6.1 UART buffer mode transmission** and **12.7 LIN Communication Automatic Baud Rate Mode** for details.

Reset input sets this register to 0000H.

Figure 12-17. Format of LIN-UARTn Buffer Control Register (UFnBUCTL) (1/2)

Address: F0258H, F0259H (UF0BUCTL), F0278H, F0279H (UF1BUCTL) After reset: 0000H R/W

	15	14	13	12	11	10	9	8
UFnBUCTL	0	0	0	0	0	0	UFnTW	UFnCON
(n = 0, 1)	7	6	5	4	3	2	1	0
	UFnECS	UFnNO	UFnRRQ	UFnTRQ	UFnBUL3	UFnBUL2	UFnBUL1	UFnBUL0

<table>
<tr><th>UFnTW</th><th>Transmission start wait bit</th></tr>
<tr><td>0</td><td>Starts transmission immediately when buffer data transmission is requested.</td></tr>
<tr><td>1</td><td>Delays starting of transmission until completion of stop bit of reception when buffer data transmission is requested.</td></tr>
<tr><td colspan="2">The UFnTW bit is used to delay starting of transmission until completion of the stop bit of reception when transmitting buffer data in LIN communication. It can be set only in automatic baud rate mode (UFnMD1, UFnMD0 = 11B). See <b>12.5.11 Transmission start wait function</b> and <b>12.7 LIN Communication Automatic Baud Rate Mode</b> for details.
<b>Cautions 1. Setting this bit is prohibited except when switching to response transmission after header reception.</b>
<b>2. The UFnTW bit becomes valid at the same time as the UFnTRQ bit is set (1).</b></td></tr>
</table>

<table>
<tr><th>UFnCON</th><th>Successive selection bit</th></tr>
<tr><td>0</td><td>The data group to be transmitted or received next is the last data group.</td></tr>
<tr><td>1</td><td>The data group to be transmitted or received next is not the last data group.
(Data transmission or reception is continued without waiting for the next header to be received.)</td></tr>
<tr><td colspan="2">The UFnCON bit indicates that the data group to be transmitted or received next is not the last data group when the multi-byte response transmission/reception function is used in LIN communication. It can be set only in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).
See <b>12.7.5 Multi-byte response transmission/reception function</b> for details.
<b>Cautions 1. Setting this bit is prohibited except when the multi-byte transmission/reception function is used.</b>
<b>2. Set the UFnCON bit at the same time as setting UFnNO, UFnRRQ, and UFnTRQ for 16-bit access.</b></td></tr>
</table>

<table>
<tr><th>UFnECS</th><th>Enhanced checksum selection bit</th></tr>
<tr><td>0</td><td>Classic checksum (used only for data byte calculation)</td></tr>
<tr><td>1</td><td>Enhanced checksum (used for calculating data byte + PID byte)</td></tr>
<tr><td colspan="2">The UFnECS bit is used to select how to handle checksum when the automatic checksum function is used in LIN communication. It is valid only when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B) and automatic checksum is enabled (UFnACE = 1).
See <b>12.7.4 Automatic checksum function</b> for details.</td></tr>
</table>

Figure 12-17. Format of LIN-UARTn Buffer Control Register (UFnBUCTL) (2/2)

UFnNO	No-response request bit
0	Response for received PID is present.
1	Response for received PID is absent.

<R> The UFnNO bit is used when a PID (PID received by a header) stored into the UFnID register is excluded in automatic baud rate mode (UFnMD1, UFnMD0 = 11B). After setting the UFnNO bit, the bit will be cleared automatically when the next BF-SF reception is complete. It can be set only in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

See **12.7 LIN Communication Automatic Baud Rate Mode** for details.

Caution Do not set the UFnTRQ and UFnRRQ bits while the UFnNO bit is “1”. Simultaneous rewriting is prohibited.

UFnRRQ	Reception request bit
0	Storing has been started/no reception request
1	Reception start request/during receive operation in automatic baud rate mode

The UFnRRQ bit is used to request starting of storing data into a buffer. It is cleared when a reception completion interrupt for the buffer is generated. It can be set only in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

See **12.7 LIN Communication Automatic Baud Rate Mode** for details.

Caution Do not set the UFnNO and UFnTRQ bits while the UFnRRQ bit is “1”. Simultaneous rewriting is prohibited.

UFnTRQ	Transmission request bit
0	Storing has been started/no transmission request
1	Transmission start request/during transmit operation when using buffer

The UFnTRQ bit is used to request starting of transmitting buffer data. It is cleared when a transmission interrupt for the data prepared in the buffer is generated. It can be set only in normal UART mode (UFnMD1, UFnMD0 = 00B) or automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

See **12.6.1 UART buffer mode transmission** and **12.7 LIN Communication Automatic Baud Rate Mode** for details.

Caution Do not set the UFnNO and UFnRRQ bits while the UFnTRQ bit is “1”. Simultaneous rewriting is prohibited.

UFnBUL3 to UFnBUL0	Buffer length bits
0	Transmits or receives 9 bytes.
1 to 9	Transmits or receives number of bytes set.
10 to 15	Transmits or receives 9 bytes.

The UFnBUL3 to UFnBUL0 bits are used to set the number of transmit or receive data in a buffer. The read value is the pointer of the current buffer. The bits are valid only in normal UART mode (UFnMD1, UFnMD0 = 00B) or automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

<R> **Caution When the auto check sum function is enabled (UFnACE=1), it does not need to include the number for a check sum in buffer length, because check sum data (1 byte) is not stored in a buffer.**

(18) Serial communication pin select register (STSEL)

The STSEL register is used to switch the input source to the timer array unit and the LIN-UARTn and CAN controller communication pins.

This register can be read or written in 1-bit units or 8-bit units.

With the 78K0R/Hx3, the STSEL register can be used to select which set of LTxD1, LRxD1 pins provided at two different ports to use.

Figure 12-18. Format of Serial Communication Pin Select Register (STSEL)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
STSEL	STSLIN1	TMCAN	STSIIC11	STSCSI00	TM30K	TMLIN1	TMLIN0	0

STSLIN1	Serial communication pin selection			
	LIN-UART pin		CAN pin	
	LTxD1	LRxD1/INTPLR1	CTxD	CRxD
0	P10	P11	P72	P73
1	P72	P73	P10	P11

TMLIN1	Control of switching channel 3 input source of timer array unit 1
0	TI13 pin input (pin input selected by using TIS1_3 bit)
1	LRxD1 pin input

TMLIN0	Control of switching channel 2 input source of timer array unit 1
0	TI12 pin input (pin input selected by using TIS1_2 bit)
1	LRxD0 pin input

Remark During LIN communication, when in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B), select the input signal of the serial data input pin (LRxDn) as a timer input by setting TMLINn =1.

(19) Port mode registers 1, 7 (PM1, PM7)

The PM1 and PM7 registers are used to set ports 1 and 7 to input or output in 1-bit units.

When using the P10/TI00/$\overline{\text{SCK10}}$/TO00/CTxD/LTxD1, P13/TI04/LTxD0/TO04, and P72/KR2/CTxD/LTxD1 pins for serial data output, clear the PM1_0, PM1_3, and PM7_2 bits to “0”, and set the output latches of P1_0, P1_3, and P7_2 to “1”.

When using the P11/TI02/SI10/LRxD1/INTPLR1/CRxD/TO02, P14/TI06/LRxD0/INTPLR0/TO06, and P73/KR3/CRxD/LRxD1/INTPLR1 pins for serial data input, set the PM1_1, PM1_4, and PM7_3 bits to “1”. At this time, the output latches of P1_1, P1_4, and P7_3 may be “0” or “1”.

The PM1 and PM7 registers can be set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remarks 1. The pins mounted depend on the product. See **1.4 Ordering Information** and **2.1 Pin Function List**.

2. See **CHAPTER 4 PORT FUNCTIONS** for port settings.

<R> **Figure 12-19. Format of Port Mode Registers 1, 7 (PM1, PM7)**

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM1_7	PM1_6	PM1_5	PM1_4	PM1_3	PM1_2	PM1_1	PM1_0

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM7_7	PM7_6	PM7_5	PM7_4	PM7_3	PM7_2	PM7_1	PM7_0

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

12.4 Interrupt Request Signals

The following three interrupt request signals are generated from LIN-UARTn.

- Status interrupt request signal (INTLSn)
- Reception complete interrupt request signal (INTLRn)
- Transmission interrupt request signal (INTLTn)

Table 12-2 shows the default priority order of these three interrupt request signals.

Table 12-2. Interrupts and Their Default Priorities

Interrupt	Default Priority
Status	Low
Reception complete	|
Transmission start/complete	High

(1) Status interrupt request signal (INTLSn)

A status interrupt request signal is generated when an error condition is detected during a reception. A UFnSTR register flag (UFnPE, UFnFE, UFnOVE, UFnDCE, UFnBSF, UFnIPE, UFnCSE, UFnRPE, UFnIDM, UFnEBD) corresponding to the detected status is set.

See **12.5.10 Status interrupt generation sources** for details.

(2) Reception complete interrupt request signal (INTLRn)

A reception complete interrupt request signal is generated when data is shifted into the receive shift register and transferred to the UFnRX register in the reception enabled status.

When a reception error occurs, a reception complete interrupt request signal is not generated, but a status interrupt request signal is generated.

No reception complete interrupt request signal is generated in the reception disabled status.

- If expansion bit operation is enabled (UFnCL = UFnEBE = 1) and expansion bit data comparison is disabled (UFnEBC = 0), a reception complete interrupt request signal is generated when the level of the inverted value set by using the expansion bit detection level select bit (UFnEBL) is detected as an expansion bit.
- When there is no error when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B) and PID reception has been completed (stop bit position), a reception complete interrupt request signal is generated.
- <R> When response reception has ended without an error when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B), a reception complete interrupt request signal is generated.

Remark n = 0, 1

(3) Transmission interrupt request signal (INTLTn)

When a transmission interrupt request is set to output upon starting a transmission (UFnITS = 0), a transmission interrupt request signal is generated when transmission from the UFnTX register to the transmit shift register has been completed.

When a transmission interrupt request is set to output upon completion of a transmission (UFnITS = 1), a transmission interrupt request signal is generated when transmitting a stop bit has been completed.

<R>

- In automatic baud rate mode (UFnMD1, UFnMD0 = 11B), if the transmitting start of the last transmitting byte of response transmission is carried out, the completion interrupt request signal of transmitting will occur.

Remark n = 0, 1

12.5 Operation

12.5.1 Data format

Full-duplex serial data reception and transmission is performed.

As shown in Figure 12-20, one data frame of transmit/receive data consists of a start bit, character bits, an expansion bit, a parity bit, and stop bits.

Specification of the character bit length within 1 data frame, parity selection, specification of the stop bit length, and specification of MSB/LSB-first transfer are performed using the UFnCTL0 register.

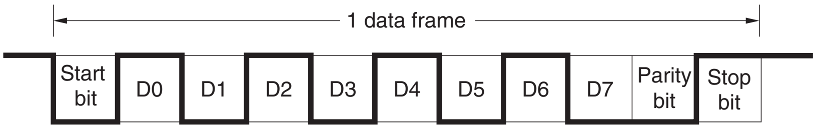
Moreover, the UFnTDL bit and UFnRDL bit of the UFnOPT0 register are used to control UART output/inverted output for the LTxDn pin and UART input/inverted input for the LRxDn pin, respectively.

Remark n = 0, 1

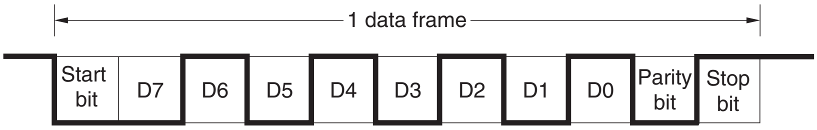
- Start bit .... 1 bit
- Character bits.......... 7 bits/8 bits
- Expansion bit .... 1 bit
- Parity bit................. Even parity/odd parity/0 parity/no parity
- Stop bit.................... 1 bit/2 bits
- <R> Output logic .... positive/inverted
- <R> Communication direction .... LSB/MSB

Figure 12-20. Format of LIN-UART Transmit/Receive Data (1/2)

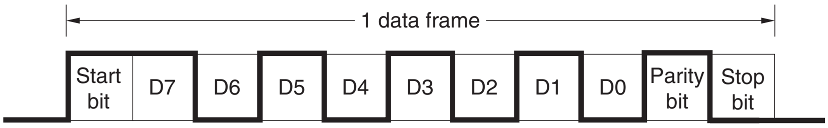
(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, LTxDn inversion



(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36H

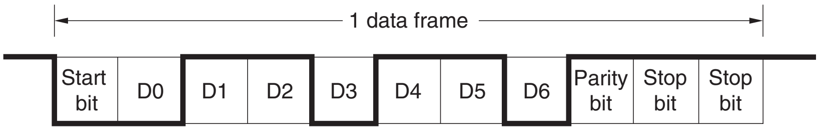
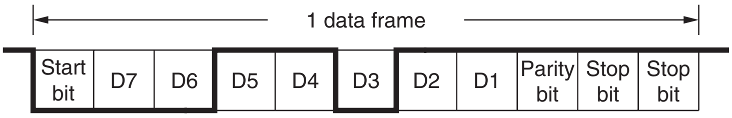
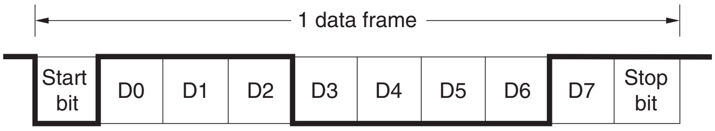


Figure 12-20. Format of LIN-UART Transmit/Receive Data (2/2)

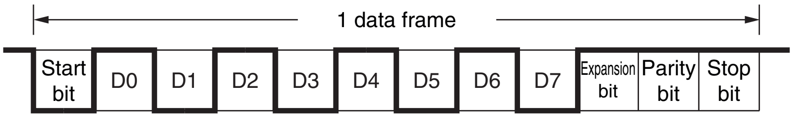
(e) **7-bit data length, MSB first, odd parity, 2 stop bits, transfer data: 36H**



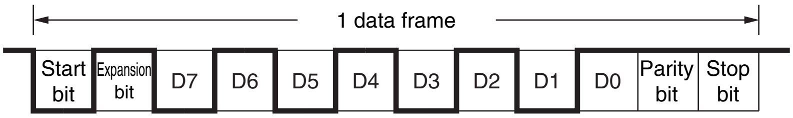
(f) **8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87H**



(g) **8-bit data length, LSB first, even parity, expansion bit: enabled, 1 stop bit, transfer data: 155H**



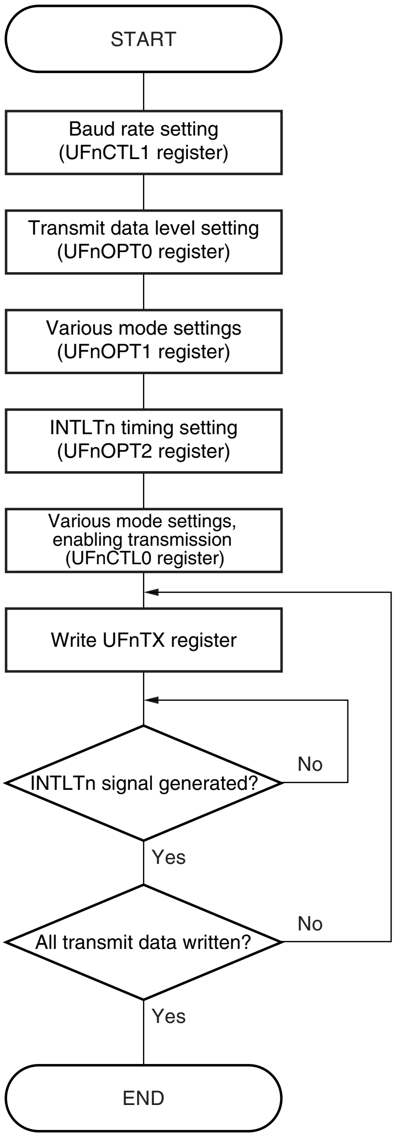
(h) **8-bit data length, MSB first, even parity, expansion bit: enabled, 1 stop bit, transfer data: 155H**



12.5.2 Data transmission

Figure 12-21 shows the procedure for transmitting data.

Figure 12-21. Transmission Processing Flow



Cautions 1. When initializing (UFnTXE = 0) the transmission unit, be sure to confirm that the transmission status flag has been reset (UFnTSF = 0). When initialization is performed while UFnTSF is "1", transmission is aborted midway.

2. During LIN communication, confirm that a status interrupt request signal (INTLSn) has been generated, because reception is performed simultaneously with transmission.

3. When data consistency error detection has been set (UFnDCS = 1) and a data consistency error has been detected during LIN communication, transmission of the next data frame or BF is stopped at the same as when a status interrupt request signal (INTLSn) is generated and a data consistency error flag is set (UFnDCE = 1).

Remarks 1. See **(2)** of **12.11 Cautions on Use for details of starting LIN-UART**.

2. n = 0, 1

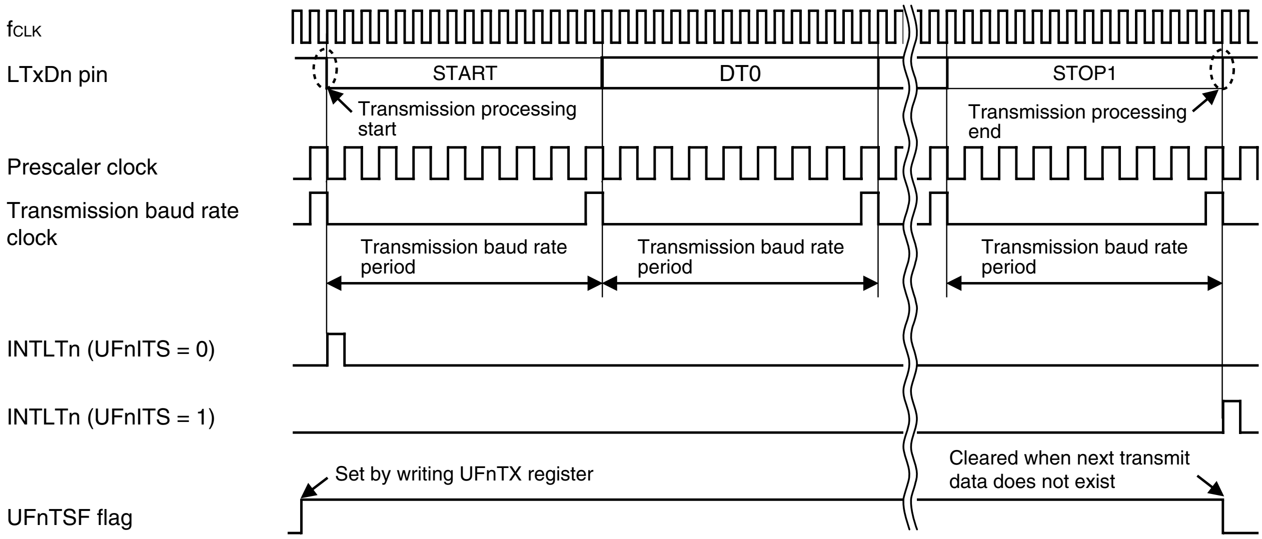
A transmission operation is started by writing transmit data to the transmit data register (UFnTX).

The data stored into the UFnTX register is transferred to the transmit shift register and a start bit, an expansion bit, a parity bit, and stop bits are added to the data, and the data are sequentially output from the LTxDn pin.

If a transmission interrupt is set upon starting a transmission (UFnITS = 0), a transmission interrupt request signal (INTLTn) is generated when transferring the data stored into the UFnTX register to the transmit shift register has been completed.

If a transmission interrupt is set upon completion of a transmission (UFnITS = 1), a transmission interrupt request signal (INTLTn) is generated when transmitting a stop bit has been completed.

Figure 12-22. Data Transmission Timing Chart



f_{CLK}
LTxDn pin
START
DT0
STOP1
Transmission processing start
Transmission processing end
Prescaler clock
Transmission baud rate clock
Transmission baud rate period
Transmission baud rate period
Transmission baud rate period
INTLTn (UFnITS = 0)
INTLTn (UFnITS = 1)
Set by writing UFnTX register
Cleared when next transmit data does not exist
UFnTSF flag

Caution **If the stop bit length is set to 2 bits (UFnSL = 1), the transmit completion interrupt (INTLTn) will be output after the second stop bit has been transmitted, at which point the transmission status flag (UFnTSF) will be cleared.**

Remark n = 0, 1

When generation of a transmission interrupt is set upon starting a transmission (UFnITS = 0), successive transmission can be performed by writing the next data to UFnTX during the transmission after INTLTn has been generated.

Figure 12-23. Diagram of Timing When Starting Successive Transmission (UFnITS = 0)

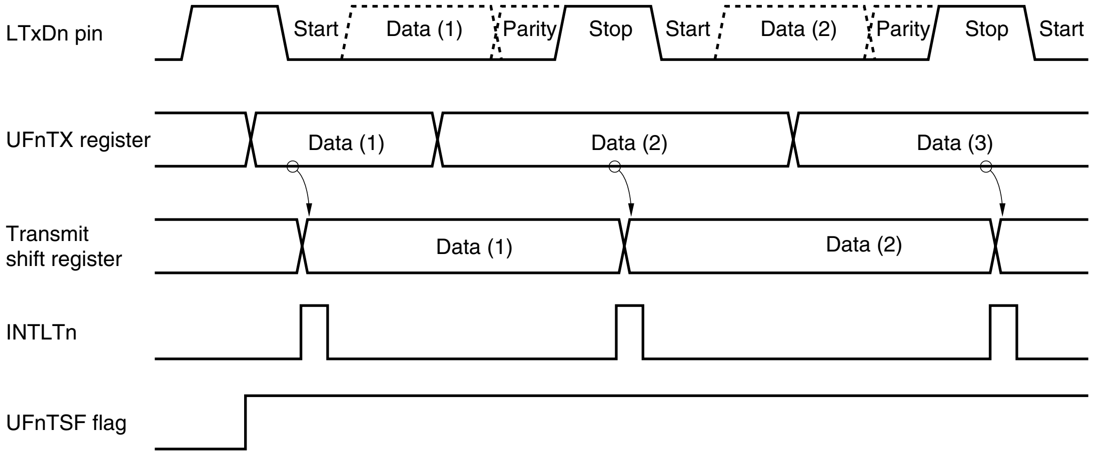
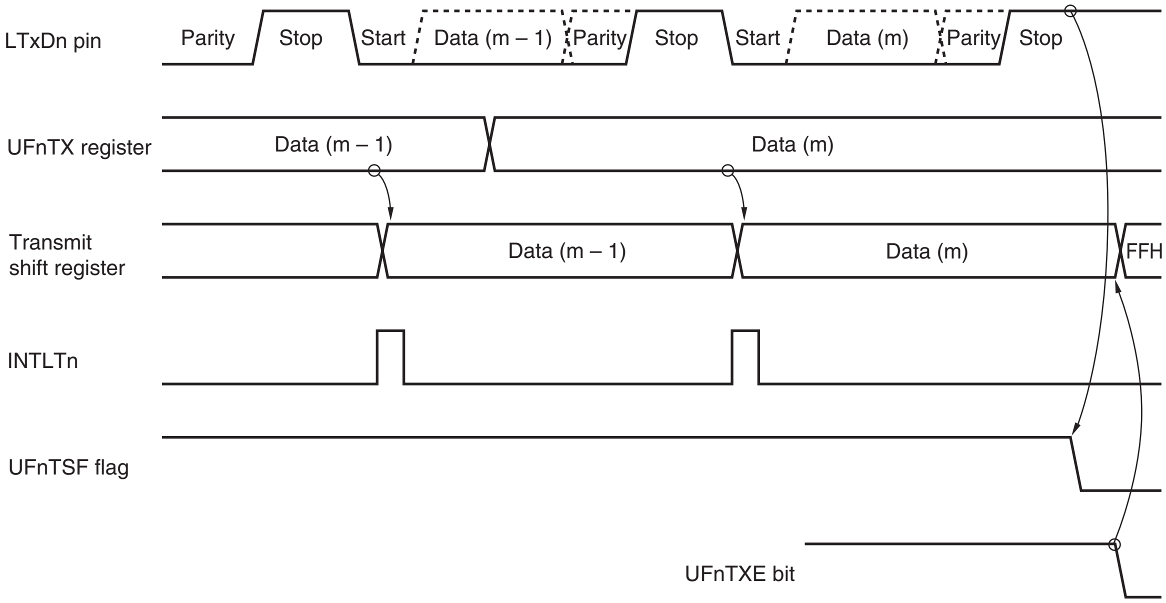


Figure 12-24. Diagram of Timing When Ending Successive Transmission (UFnITS = 0)

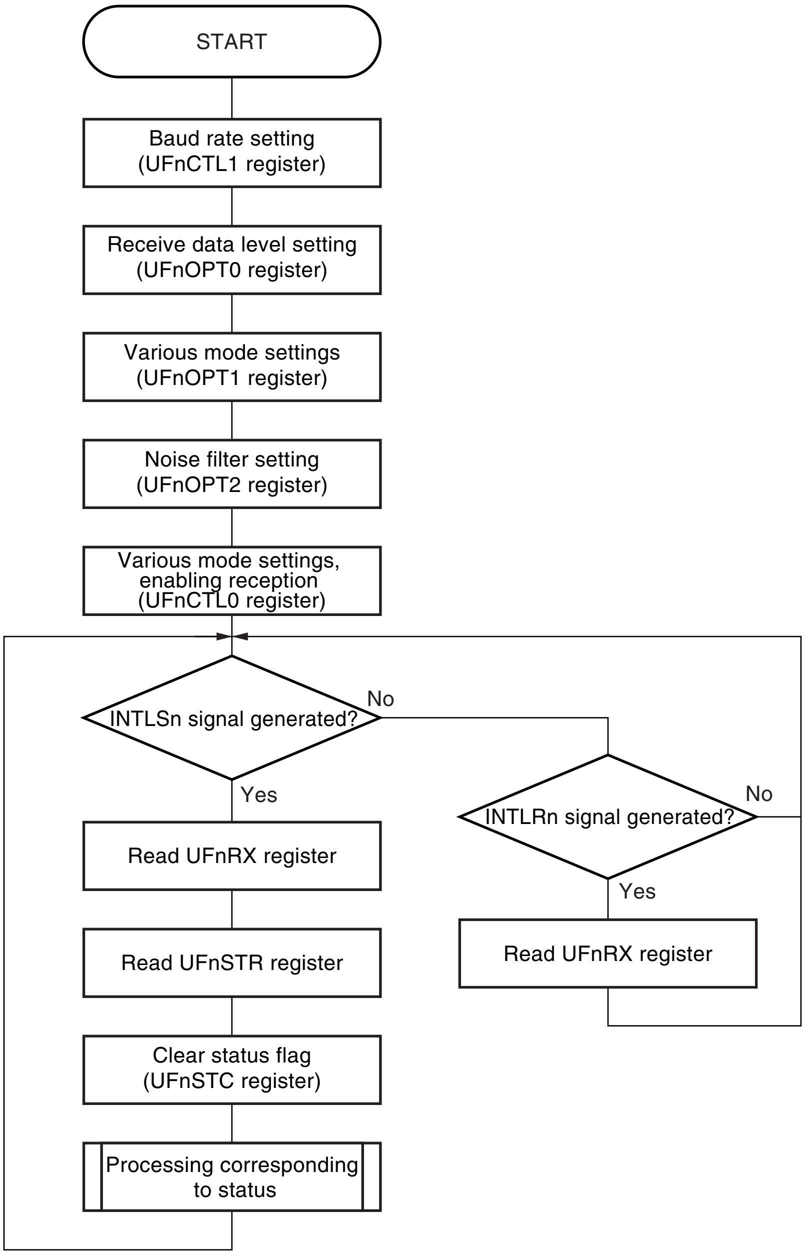


Remark n = 0, 1

12.5.3 Data reception

Figure 12-25 shows the procedure for receiving data.

Figure 12-25. Reception Processing Flow



Cautions 1. When initializing (UFnRXE = 0) the reception unit, be sure to confirm that the reception status flag has been reset (UFnRSF = 0). When initialization is performed while UFnRSF is “1”, reception is aborted midway.

2. Be sure to read the receive data register (UFnRX) when a reception error has occurred.
If the UFnRX register is not read, an overrun error occurs upon completion of receiving the next data.

Remarks 1. See **(2)** of **12.11 Cautions on Use for details of starting LIN-UART**.

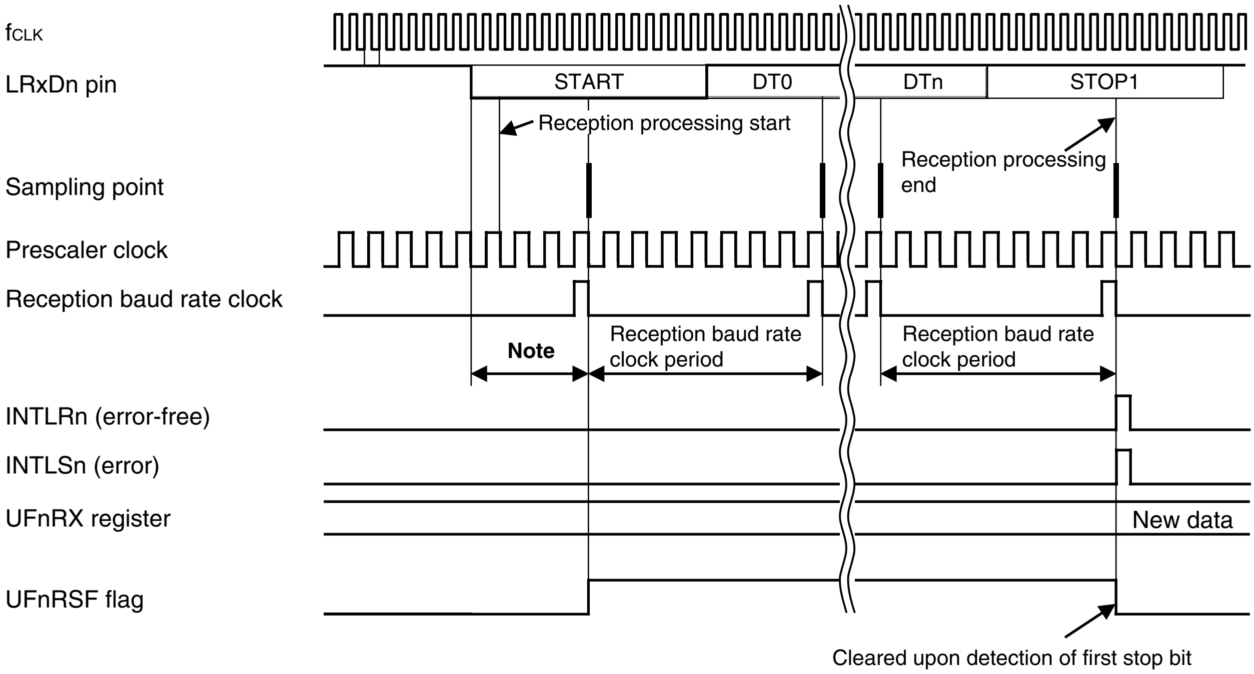
2. n = 0, 1

When the LRxDn pin is sampled by using the operating clock and a falling edge is detected, data sampling of the LRxDn pin is started and is recognized as a start bit if it is at low level at a timing of half the reception baud rate clock period after the falling edge has been detected. When the start bit has been recognized, a reception operation is started and serial data is sequentially stored into the receive shift register according to the baud rate set. When a stop bit has been received, the data stored into the receive shift register is transferred to the receive data register (UFnRX) at the same time a reception complete interrupt request signal (INTLRn) is generated.

When an overrun error has occurred (UFnOVE = 1), however, the receive data is not transferred to the UFnRX register but discarded. When any other error has occurred, the reception is continued up to the reception position of the stop bit and the receive data is transferred to the UFnRX register.

After the occurrence of any reception error, INTLSn is generated after completion of the reception and INTLRn is not generated.

Figure 12-26. Data Reception Timing Chart


fCLK
LRxDn pin
START
DT0
DTn
STOP1
Reception processing start
Reception processing end
Sampling point
Prescaler clock
Reception baud rate clock
Note
Reception baud rate clock period
Reception baud rate clock period
INTLRn (error-free)
INTLSn (error)
UFnRX register
New data
UFnRSF flag
Cleared upon detection of first stop bit

Note One-half the reception baud rate clock period

Cautions 1. The start bit is not recognized when a high level is detected at a timing of half the reception baud rate clock period after the falling edge of the LRxDn pin was detected.

**2. A reception always operates with the number of stop bits as 1.
At that time, the second stop bit is ignored.**

3. When a low level is constantly input to the LRxDn pin before an operation to enable reception is performed, the receive data is not identified as a start bit.

4. For successive reception, the next start bit can be detected immediately after a stop bit of the first receive data has been detected (upon generation of a reception complete interrupt).

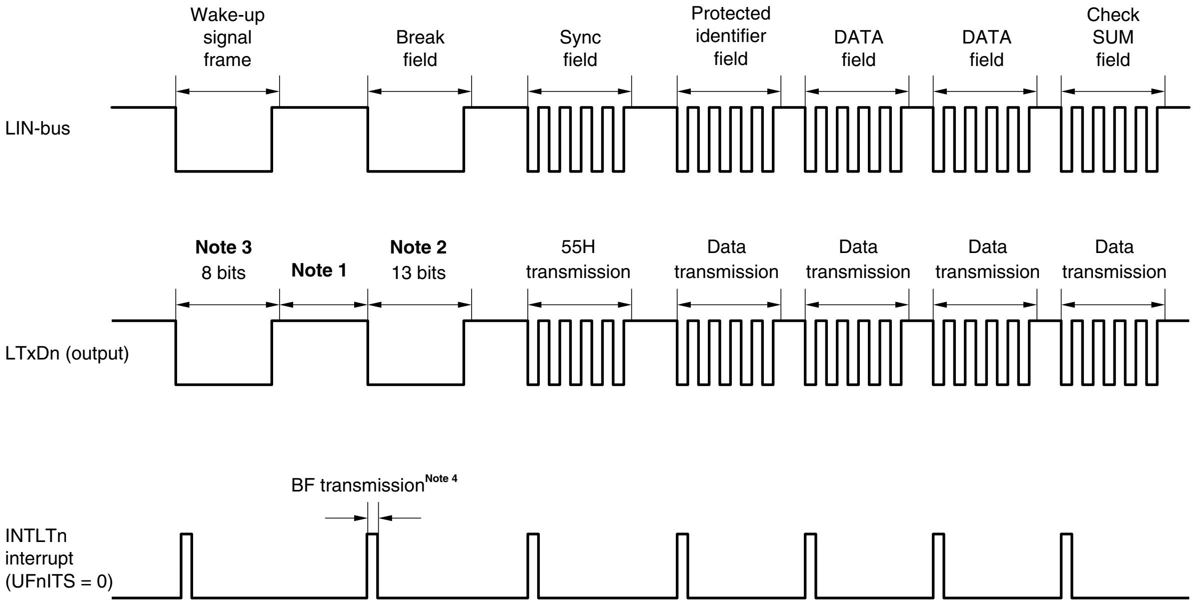
5. Be sure to enable reception (UFnRXE = 1) after having changed the UFnRDL bit. If the UFnRDL bit is changed after having enabled reception, the start bit may be detected falsely.

Remark n = 0, 1

12.5.4 BF transmission/reception format

The 78K0R/Hx3 has a BF (Break Field) transmission/reception control function to enable use of the LIN (Local Interconnect Network) function.

Figure 12-27. LIN Transmission Manipulation Outline



Notes 1. The interval between each field is controlled by software.

2. BF output is performed by hardware. The output width is the bit length set by the UFnBLS2 to UFnBLS0 bits of the UFnOPT0 register. If even finer output width adjustments are required, such adjustments can be performed using the UFnBRS11 to UFnBRS0 bits of the UFnCTL1 register.

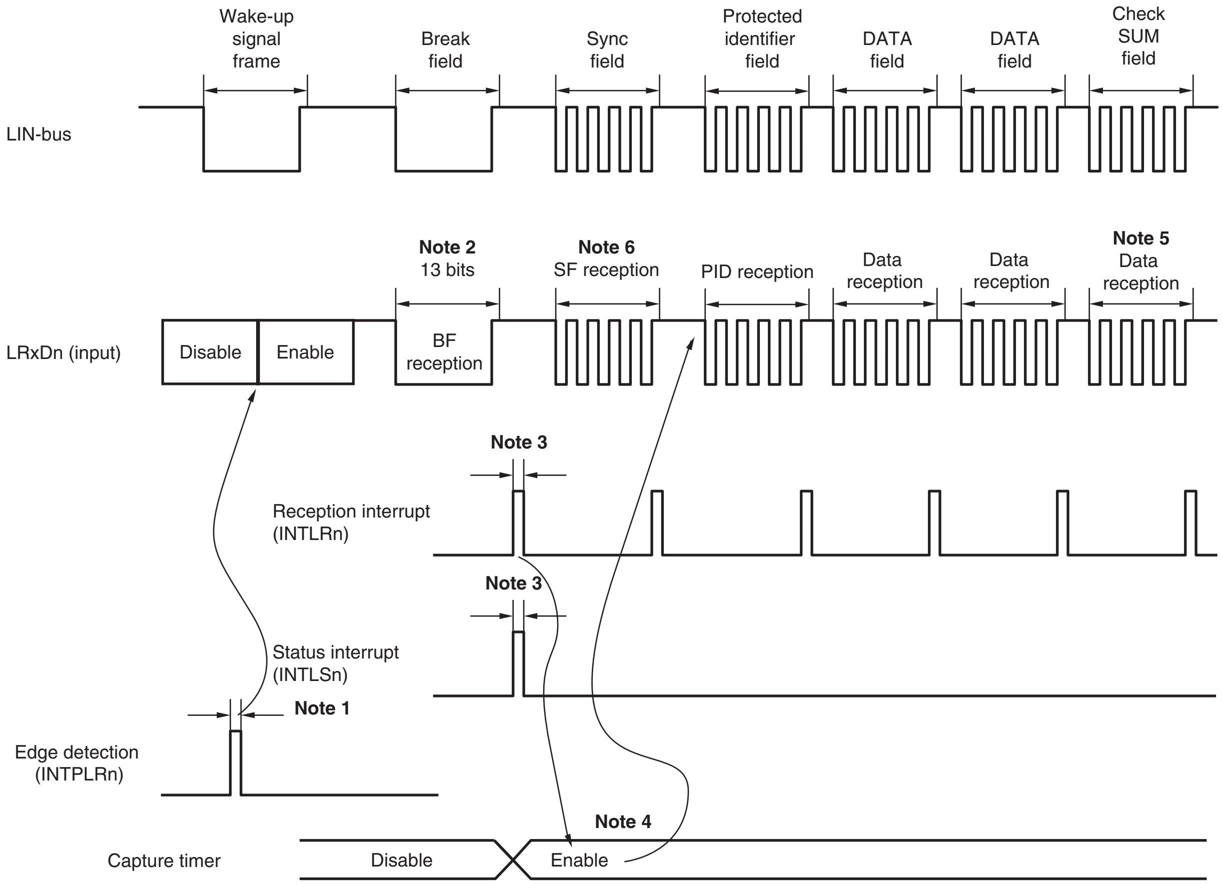
3. 80H transfer in the 8-bit mode or BF transmission is substituted for the wakeup signal.

4. A transmission enable interrupt request signal (INTLTn) is output at the start of each transmission. The INTLTn signal is also output at the start of each BF transmission. Be sure to clear UFnOPT2.UFnITS to “0” when starting a transmission, so that a transmission interrupt is generated.

Remarks 1. Figure 12-27 shows the LIN transmission manipulation outline when in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B). See **12.7 LIN Communication Automatic Baud Rate Mode** for when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

2. n = 0, 1

Figure 12-28. LIN Reception Manipulation Outline



Notes 1. After enabling LIN-UARTn after the wakeup signal reception and permitting reception operation, BF receiving trigger bit is set if needed.

2. If a BF reception of at least 11 bits is detected, the BF reception is judged to be ended normally.

<R> **3.** When BF reception has ended normally, In the normal UART Mohd (UFnMD1, UFnMD0 = 00B), the completion interrupt request signal (UFTIR) of reception occurs.

The status interrupt request signal (INTLSn) is generated in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B), and a successful BF reception flag (UFnBSF) is set. When the BF reception flag (UFnBRF) is “1”, detection of overrun, parity, and framing errors (UFnOVE, UFnPE, UFnFE) is not performed during BF reception.

4. Connect the LRxDn pin to the TI (capture input) of the timer array unit. Enable the timer by using a BF reception complete interrupt, measure the baud rate from the SF transfer data, and calculate the baud rate error. Set a reception state by stopping the LIN-UARTn reception operation after SF reception and re-setting the value of LIN-UARTn control register 1 (UFnCTL1) obtained by correcting the baud rate error.

5. Classification of a checksum field is performed by using software. The processing that initializes LIN-UARTn after CSF reception and sets to a successful BF reception wait state (UFnBRF = 1) again is also performed by using software. In BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B), however, BF reception can be automatically performed without setting to a successful BF reception wait state (UFnBRF = 1) again.

<R> **6.** Synch Field calculates the baud rate of transmission by using the capture function of TAU. When stopping interruption from UART, reception interruption can be stopped by carrying out reception to a stop.

Remarks 1. See **12.7 LIN Communication Automatic Baud Rate Mode** for when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

2. n = 0, 1

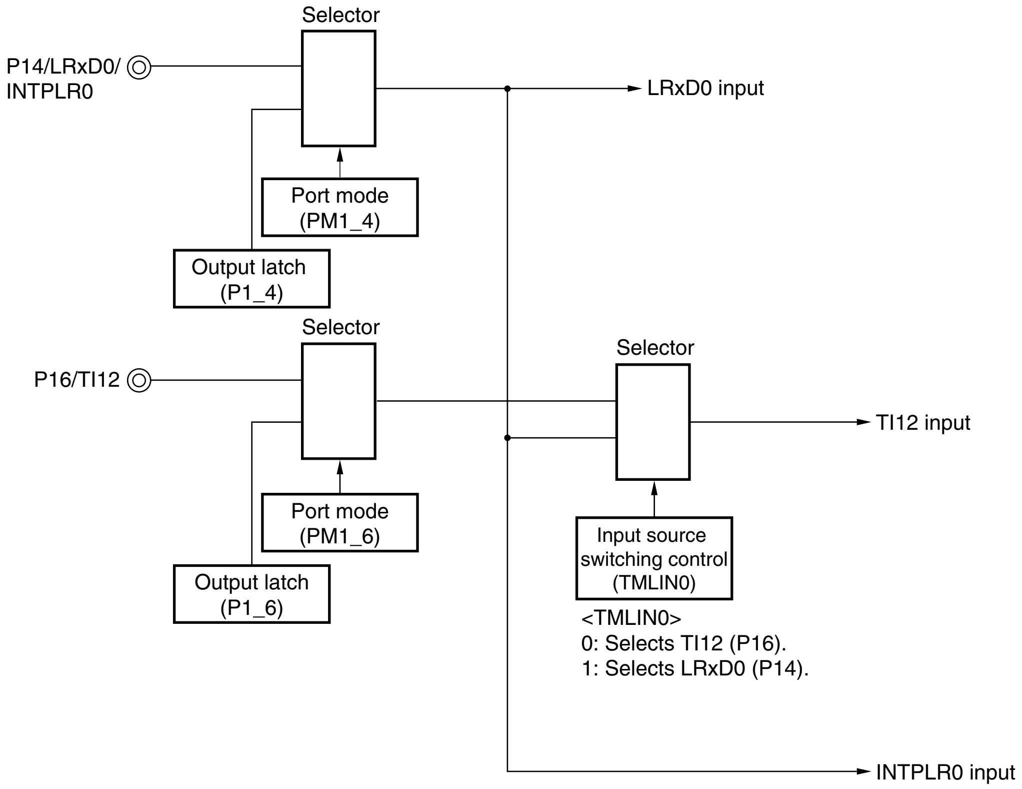
Figures 12-29 to 12-32 show the port configurations for LIN reception manipulation.

Wakeup signals transmitted from the LIN master are received via INTPLRn edge detection. The baud rate error can be calculated by measuring the length of a sync field transmitted from the LIN master via an external event capture operation of the timer array unit (TAU).

The input source of the reception port input (LRxDn) can be input to the LIN-UARTn reception pin interrupt (INTPLRn) and timer array unit (TAU) via serial communication pin selection (TMLINn), without connecting them externally.

(1) LIN-UART0

Figure 12-29. Port Configuration of LIN Reception Manipulation (78K0R/HC3, 78K0R/HE3)



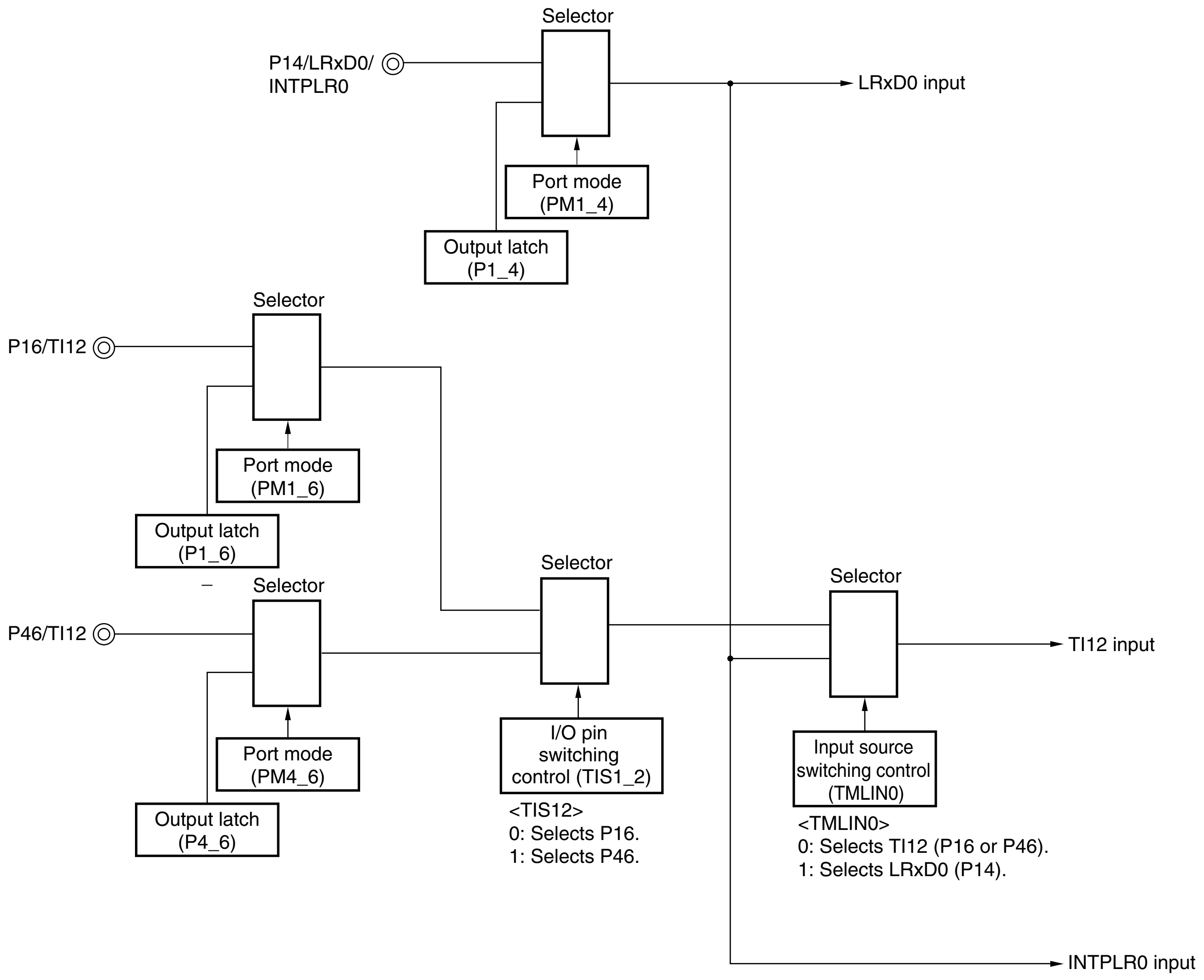
Remark TMLIN0: Bit 1 of the serial communication pin select register (STSEL) (see **Figure 12-18**)

A summary of the peripheral functions to be used in LIN communication operation is given below.

<Peripheral functions to be used>

- LIN-UART0 reception pin interrupt (INTPLR0); Wakeup signal detection
  Purpose: Detecting wakeup signal edges and detecting the start of communication
- Channel 12 (TI12) of the timer array unit (TAU); Baud rate error detection
  Purpose: Detecting the length of a sync field (SF) and detecting the baud rate error by dividing the sync field length by the number of bits (measuring the intervals of TI12 input edges in capture mode)
- Asynchronous serial interface LIN-UART0

Figure 12-30. Port Configuration of LIN Reception Manipulation (78K0R/HF3, 78K0R/HG3)



Remark TIS1_2: Bit 2 of timer input select register 1 (TIS1) (see **CHAPTER 6 TIMER ARRAY UNIT**)
TMLIN0: Bit 1 of the serial communication pin select register (STSEL) (see **Figure 12-18**)

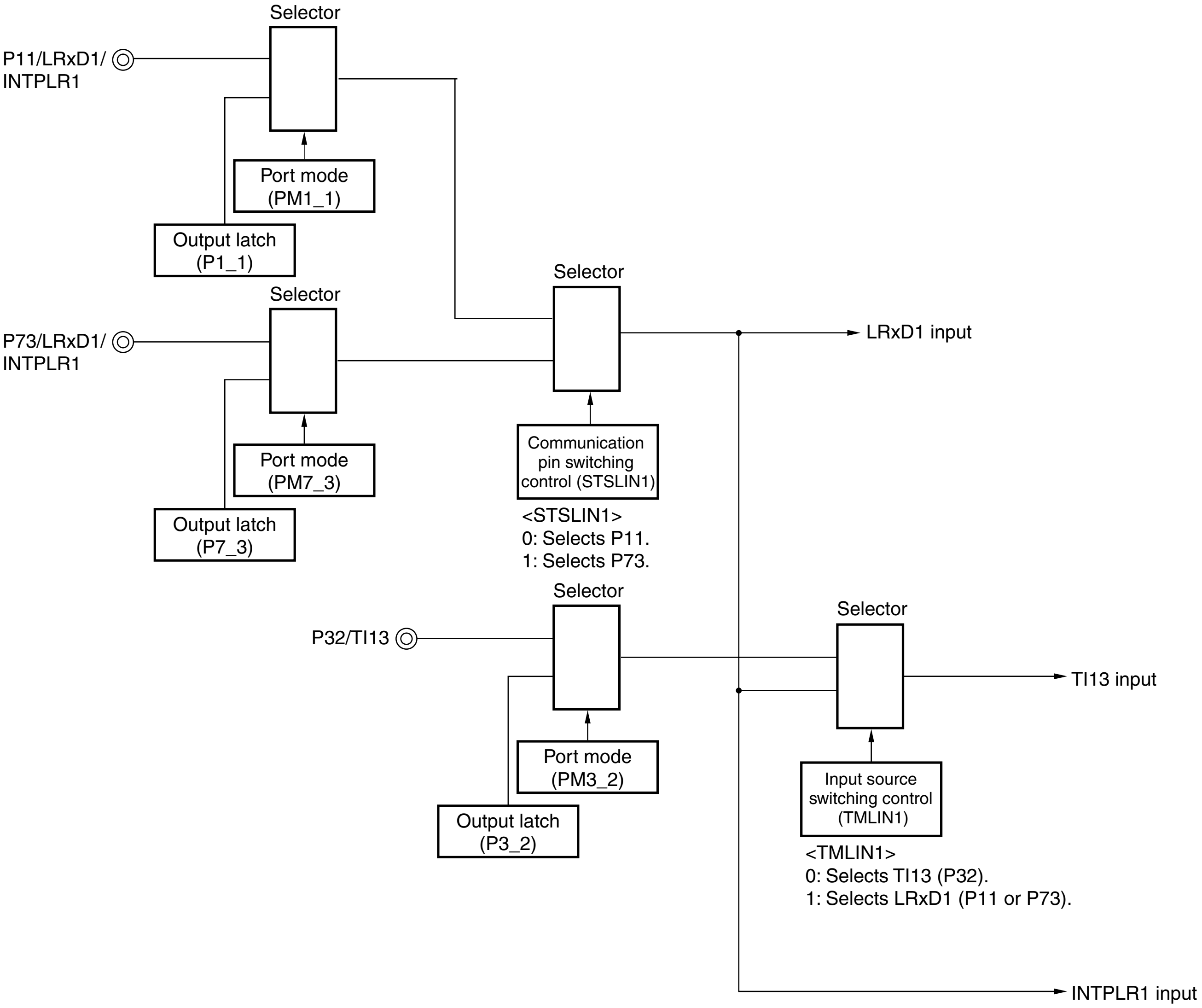
A summary of the peripheral functions to be used in LIN communication operation is given below.

<Peripheral functions to be used>

- LIN-UART0 reception pin interrupt (INTPLR0); Wakeup signal detection
  Purpose: Detecting wakeup signal edges and detecting the start of communication
- Channel 12 (TI12) of the timer array unit (TAU); Baud rate error detection
  Purpose: Detecting the length of a sync field (SF) and detecting the baud rate error by dividing the sync field length by the number of bits (measuring the intervals of TI12 input edges in capture mode)
- Asynchronous serial interface LIN-UART0

(2) LIN-UART1

Figure 12-31. Port Configuration of LIN Reception Manipulation (78K0R/HC3, 78K0R/HE3)



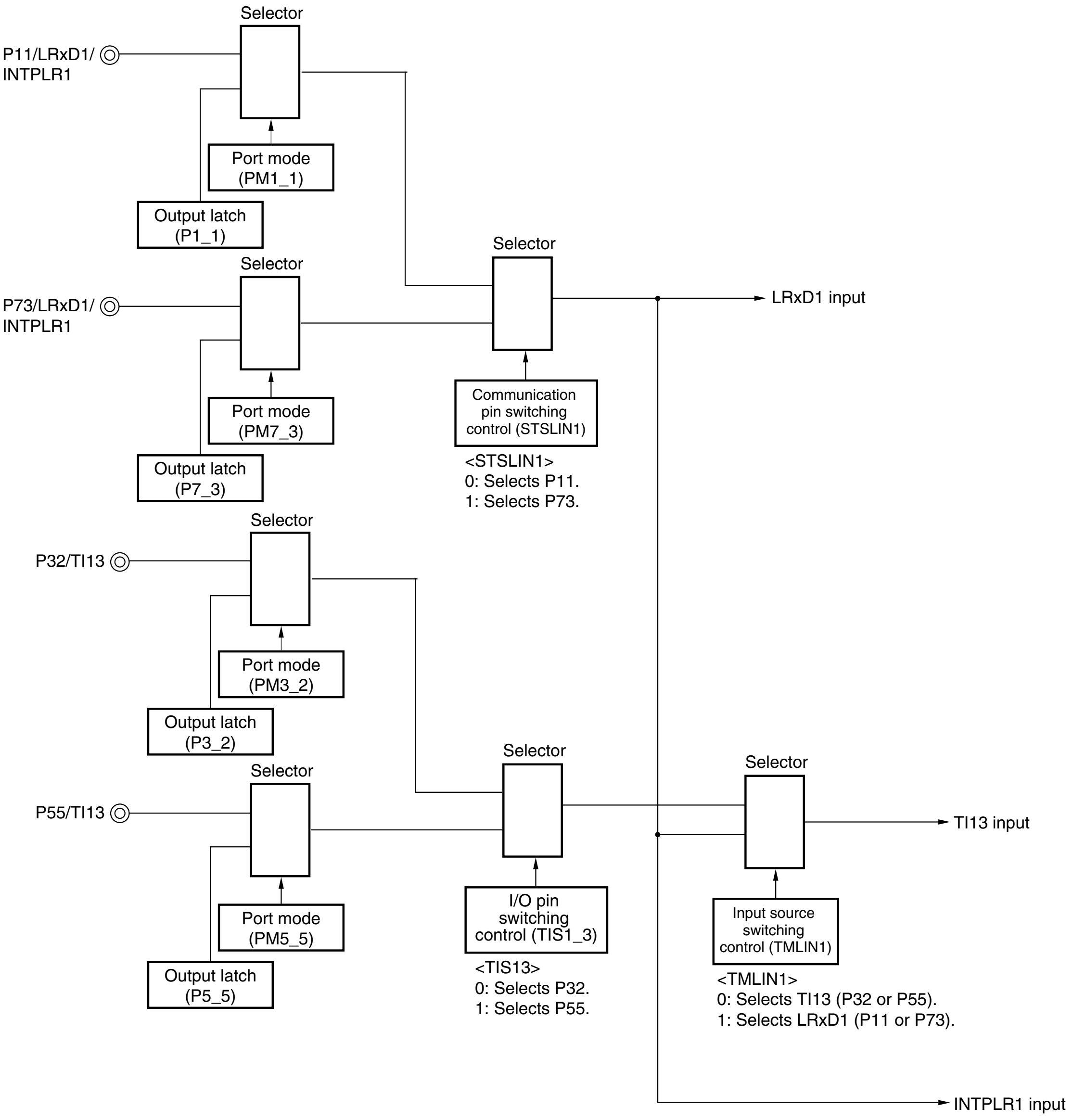
Remark STSLIN1: Bit 7 of the serial communication pin select register (STSEL) (see **Figure 12-18**)
TMLIN1: Bit 2 of the serial communication pin select register (STSEL) (see **Figure 12-18**)

A summary of the peripheral functions to be used in LIN communication operation is given below.

<Peripheral functions to be used>

- LIN-UART1 reception pin interrupt (INTPLR1); Wakeup signal detection
  Purpose: Detecting wakeup signal edges and detecting the start of communication
- Channel 13 (TI13) of the timer array unit (TAU); Baud rate error detection
  Purpose: Detecting the length of a sync field (SF) and detecting the baud rate error by dividing the sync field length by the number of bits (measuring the intervals of TI13 input edges in capture mode)
- Asynchronous serial interface LIN-UART1

Figure 12-32. Port Configuration of LIN Reception Manipulation (78K0R/HF3, 78K0R/HG3)



Remark		
	STSLIN1:	Bit 7 of the serial communication pin select register (STSEL) (see **Figure 12-18**)
	TIS1_3:	Bit 3 of timer input select register 1 (TIS1) (see **CHAPTER 6 TIMER ARRAY UNIT**)
	TMLIN1:	Bit 2 of the serial communication pin select register (STSEL) (see **Figure 12-18**)

A summary of the peripheral functions to be used in LIN communication operation is given below.

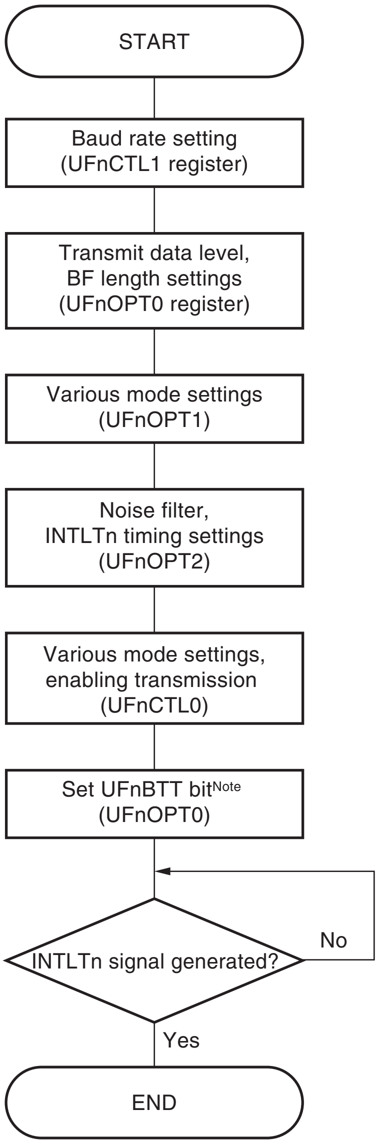
<Peripheral functions to be used>

- LIN-UART1 reception pin interrupt (INTPLR1); Wakeup signal detection
  Purpose: Detecting wakeup signal edges and detecting the start of communication
- Channel 13 (TI13) of the timer array unit (TAU); Baud rate error detection
  Purpose: Detecting the length of a sync field (SF) and detecting the baud rate error by dividing the sync field length by the number of bits (measuring the intervals of TI13 input edges in capture mode)
- Asynchronous serial interface LIN-UART1

12.5.5 BF transmission

Figure 12-33 describes the processing of BF transmission in LIN communication.

Figure 12-33. BF Transmission Processing Flow



<R> **Note** In the case of Normal UART mode (UFnMD1, UFnMD0 = 00B), please set a UFnBRT bit simultaneously.

Caution Set the following values when performing BF transmission.
The transmit data level is normal output (UFnTDL = 0).
Communication direction control is LSB first (UFnDIR = 1).
The parity selection bit is no parity bit output (UFnPS1, UFnPS0 = 00B).
The data character length is 8 bits (UFnCL = 1).
Transmission interrupt is when starting transmission (UFnITS = 0).

Remarks 1. See **(2)** of **12.11 Cautions on Use for details of starting LIN-UART**.

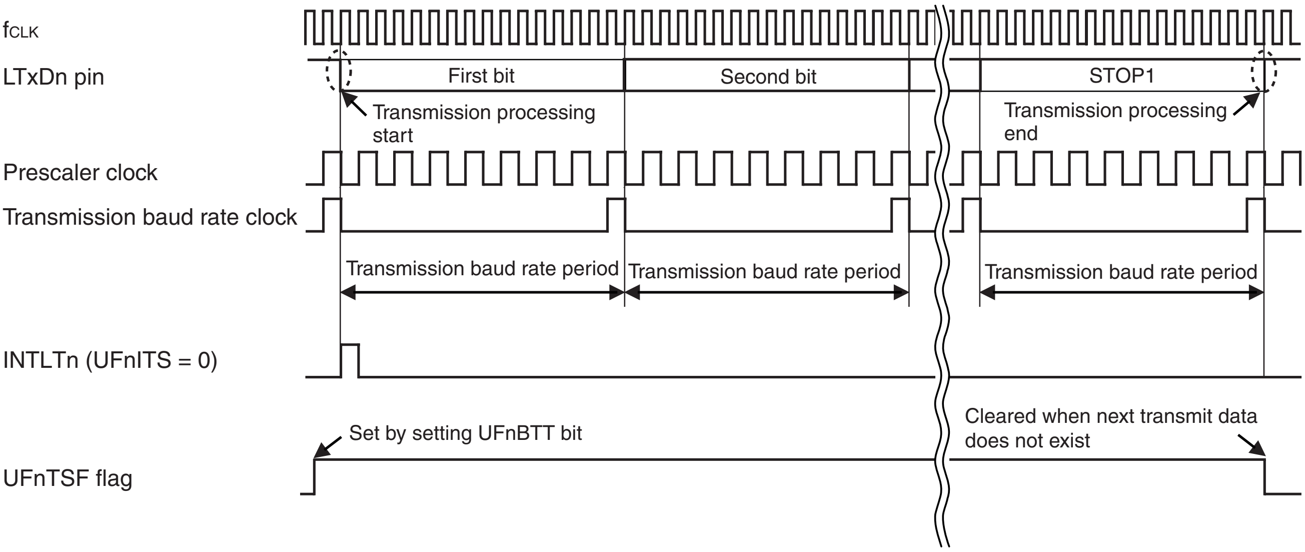
2. n = 0, 1

A BF transmission operation is started when a BF transmission trigger (UFnBTT) is set. A low level of bits 13 to 20 specified by the BF length selection bits (UFnBLS2 to UFnBLS0) is output to the LTxDn pin. A transmission interrupt request signal (INTLTn) is generated when the BF transmission is started. After the BF transmission ends, the BF transmission state is automatically released and operation is returned to normal UART transmission mode.

The transmission operation stays in a wait state until the data to be transmitted is written to the UFnTX register, or a BF transmission trigger (UFnBTT) is set. Start the next transmission operation after having confirmed that the BF has been received normally according to the reception complete interrupt (INTLRn) during the BF transmission or the status interrupt (INTLSn).

Figure 12-34. BF Transmission Timing Example

<R>


fCLK
LTxDn pin
First bit
Second bit
STOP1
Transmission processing start
Transmission processing end
Prescaler clock
Transmission baud rate clock
Transmission baud rate period
Transmission baud rate period
Transmission baud rate period
INTLTn (UFnITS = 0)
Set by setting UFnBTT bit
Cleared when next transmit data does not exist
UFnTSF flag

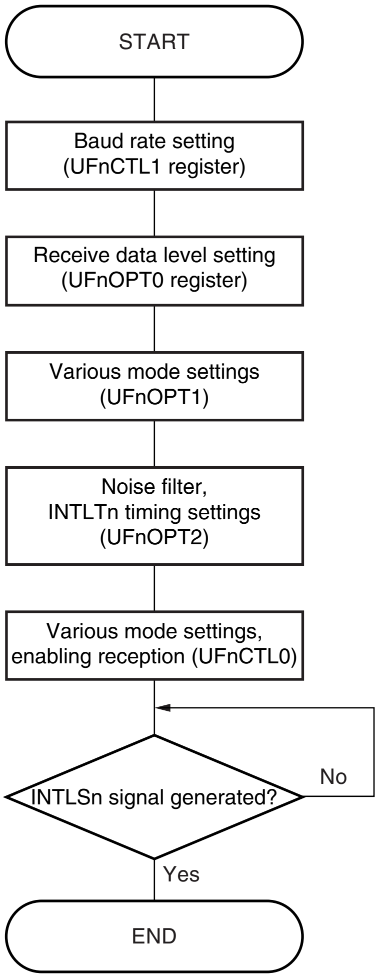
Caution **When the stop bit length is set to 2 bits (UFnSL = 1), the transmission status flag (UFnTSF) is cleared when the second stop bit has been completed.**

Remark n = 0, 1

12.5.6 BF reception

Figure 12-35 describes the processing of BF reception in LIN communication.

Figure 12-35. BF Reception Processing Flow



Caution **Set the following values when performing BF transmission.**
The input logic level is normal input (UFnRDL = 0).
Communication direction control is LSB first (UFnDIR = 1).
The parity selection bit is no parity bit output (UFnPS1, UFnPS0 = 00B).
The data character length is 8 bits (UFnCL = 1).
Transmission interrupt is when starting transmission (UFnITS = 0).
BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B) as the mode.

Remarks 1. Figure 12-35 shows the reception processing flow of LIN communication in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B).
See **12.7 LIN Communication Automatic Baud Rate Mode** for when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

2. See **(2)** of **12.11 Cautions on Use for details of starting LIN-UART**.

3. n = 0, 1

<R> When the falling edge of the LRxDn input level is detected, the BF length is measured by counting up the internal counter until a rising edge is detected.

If the BF length is at least 11 bits when a rising edge has been detected, BF reception is judged as being normal, BF reception ends.

When ending BF reception, a successful BF reception flag (UFnBSF) is set at the same time as a status interrupt request signal (INTLSn) is generated.

Detection of overrun, parity, and framing (UFnOVE, UFnPE, UFnFE) errors is controlled. Moreover, data transfer from

the receive shift register to the receive data register (UFnRX) is not performed. BF reception is judged as being abnormal if the BF width is less than 11 bits.

In that case, error status (UFnSTR) is set at the same time it generates a status interrupt request signal (INTLSn).

When performing a transmission for which a data consistency check is enabled (UFnDCS = 1), a data consistency error flag (UFnDCE) is set and a status interrupt request signal (INTLSn) is output when a mismatch between the transmit data and receive data is detected, regardless of whether BF reception is performed successfully or fails. At that time, INTLRn is not output.

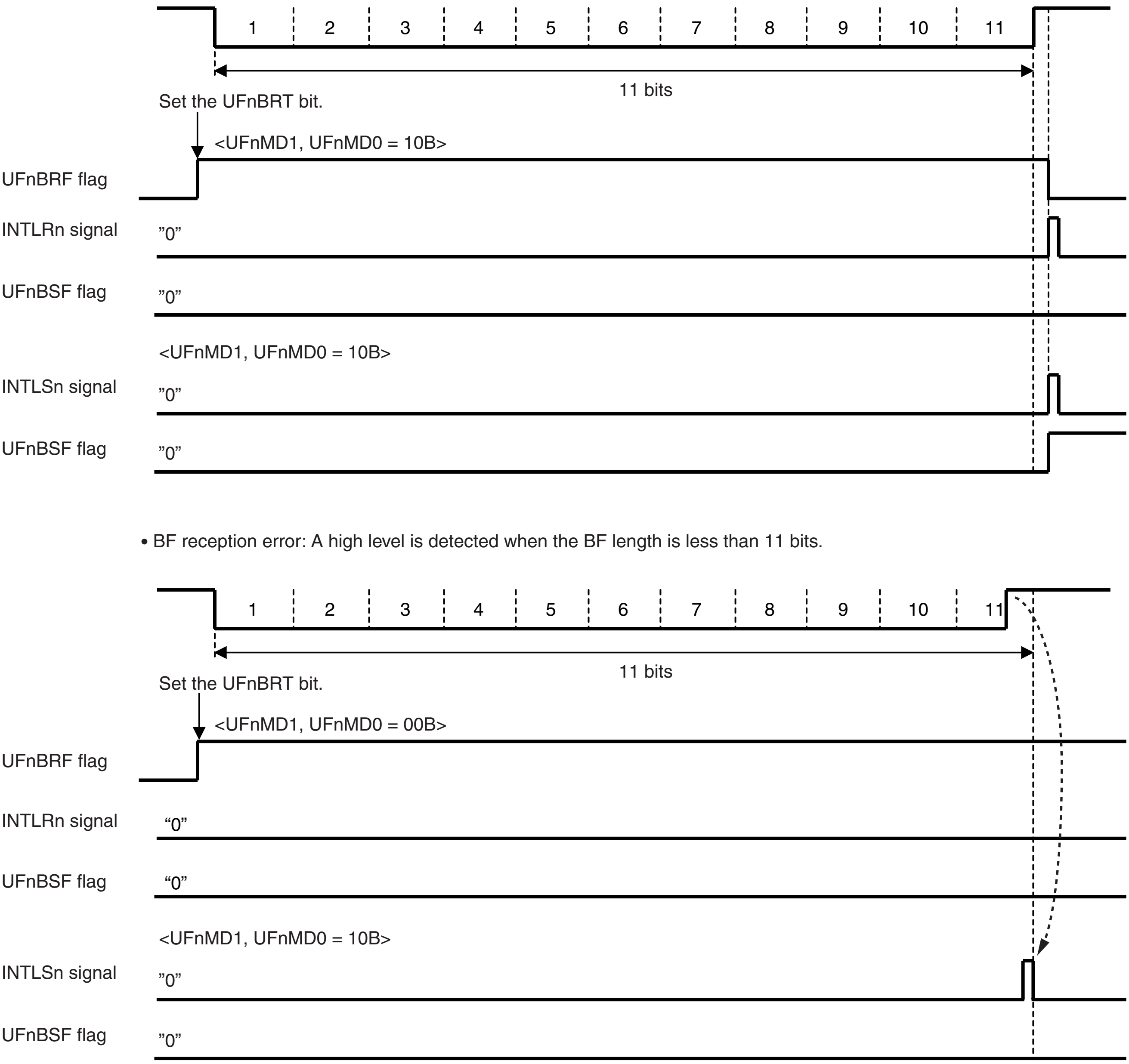
When it is in the BF reception enable mode (UFnMD1, UFnMD0 = 10B) or automatic baud rate mode (UFnMD1, UFnMD0 = 11B) during communication, LIN-UART can detect new BF reception also in data communications.

Remark n = 0, 1

<R>

Figure 12-36. BF Reception Timing Example

• Normal BF reception: A high level is detected after the BF length has exceeded 11 bits.



• BF reception error: A high level is detected when the BF length is less than 11 bits.

Caution **The UFnBRF bit is set by setting the UFnBRT bit to "1" and cleared upon normal BF reception. It is the same operation during communication also on the BF reception enable mode(UFnMD1, UFnMD0 = 10B).**

Remark n = 0, 1

12.5.7 Parity types and operations

Caution When using the LIN communication, fix the UFnPS1 and UFnPS0 bits of the UFnCTL0 register to 00 (n = 0, 1).

The parity bit is used to detect bit errors in the communication data. Normally the same parity bit is used on the transmission side and the reception side.

In the case of even parity and odd parity, it is possible to detect 1-bit (odd-count) errors. In the case of 0 parity and no parity, errors cannot be detected.

(1) Even parity

(a) During transmission

The number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so as to be an even number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 1
- Even number of bits whose value is “1” among transmit data: 0

(b) During reception

The number of bits whose value is “1” among the reception data, including the parity bit, is counted, and if it is an odd number, a parity error is output.

(2) Odd parity

(a) During transmission

Opposite to even parity, the number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so that it is an odd number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 0
- Even number of bits whose value is “1” among transmit data: 1

(b) During reception

The number of bits whose value is “1” among the receive data, including the parity bit, is counted, and if it is an even number, a parity error is output.

(3) 0 parity

During transmission, the parity bit is always made 0, regardless of the transmit data.

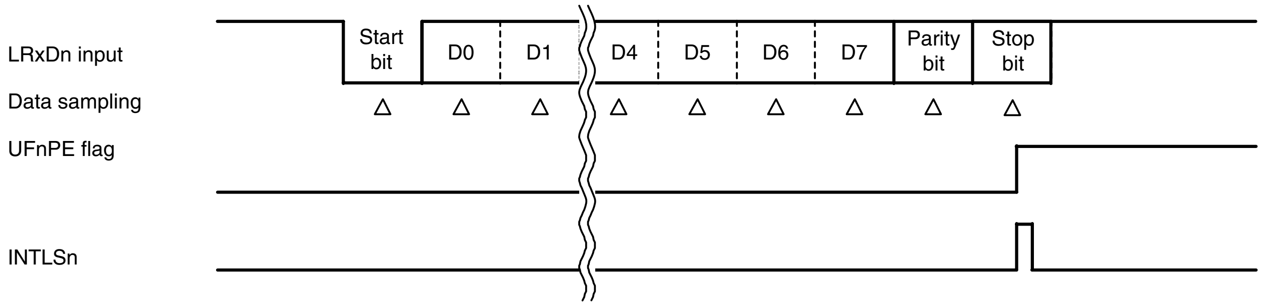
During reception, parity bit check is not performed. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(4) No parity

No parity bit is added to the transmit data.

Reception is performed assuming that there is no parity bit. No parity error occurs since there is no parity bit.

Figure 12-37. Parity Error Occurrence Timing



12.5.8 Data consistency check

When the data consistency check selection bit (UFnDCS) is set to “1”, transmit data and receive data are compared during transmission operation, even if the reception enable bit is disabled (UFnRXE = 0).

When reception is enabled (UFnRXE = 1), it is also checked that reception processing is not ended early during transmission processing.

When either a mismatch between transmission and reception signals or an early end of reception processing is detected during transmission processing, operation is judged as being abnormal, a status interrupt request signal (INTLSn) is output, and a data consistency error flag (UFnDCE) is set. Even if the next transmit data has already been written to the transmit data register (UFnTX), the next transmission is not performed. (The written data within UFnTX is ignored.) When the BF transmission trigger bit (UFnBTT) has been set, a BF is not transmitted.

To restart transmission, transmit data must be written to the transmit data register (UFnTX) or the BF transmission trigger bit (UFnBTT) set after the end of transmission has been confirmed (UFnTSF = 0) and the data consistency error <R> flag (UFnDCE) or the LINnEN bit of the PER0 register has been cleared and has been set again. When a buffer is used, communication is stopped even if data not transferred remains in the buffer.

<R> **Caution A store operation of receive data is not affected by whether a data consistency error exists. Storing is performed even if a consistency error occurs.**
In order to avoid an overrun error, please read receiving data at the time of UFnRXE = 1.
However, in a setup of UFnRXE=0, since reception operation is not carried out and receiving data is not stored, it is not necessary to read receiving data.

Remark n = 0, 1

(1) Mismatch between transmission and reception signals

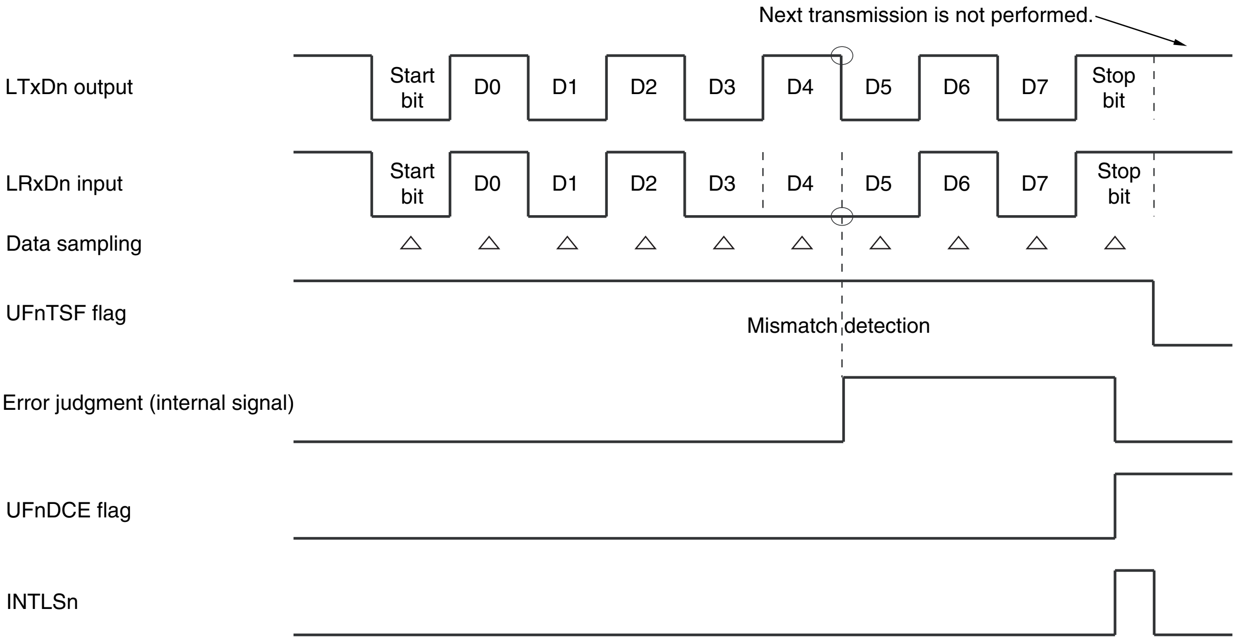
Serial transmission and reception signals are compared during data (or BF) transmission, a detected mismatch is judged as being abnormal, and the UFnDCE bit is set (1) at the same time a status interrupt (INTLSn) is generated.

During data transmission, the comparison is performed from the start bit to the first stop bit.

During BF transmission, the comparison is performed from the first bit of the BF to the first stop bit.

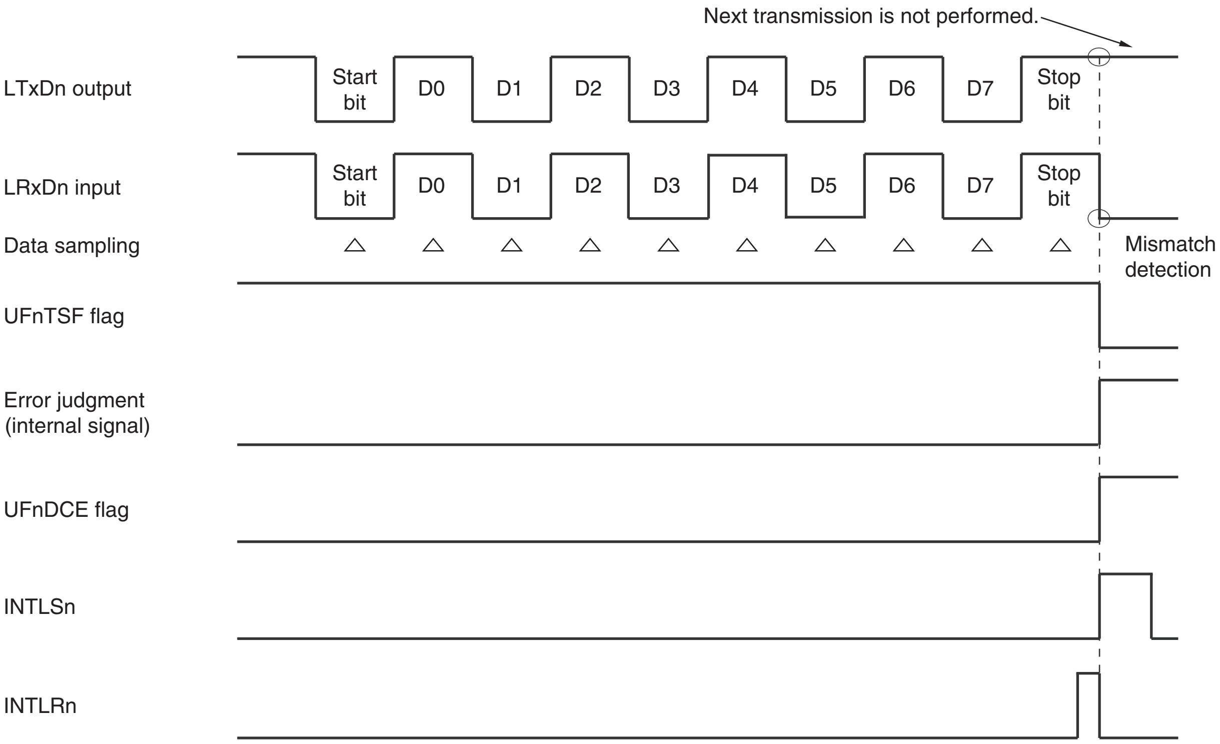
A consistency check is not performed for the second stop bit, even if the stop bit length is specified as two bits by using the stop bit length select bit (UFnSL).

Figure 12-38. Data Consistency Error Occurrence Timing Example 1 (UFnBRF = 0)



Remark n = 0, 1

Figure 12-39. Data Consistency Error Occurrence Timing Example 2 (UFnBRF = 0)

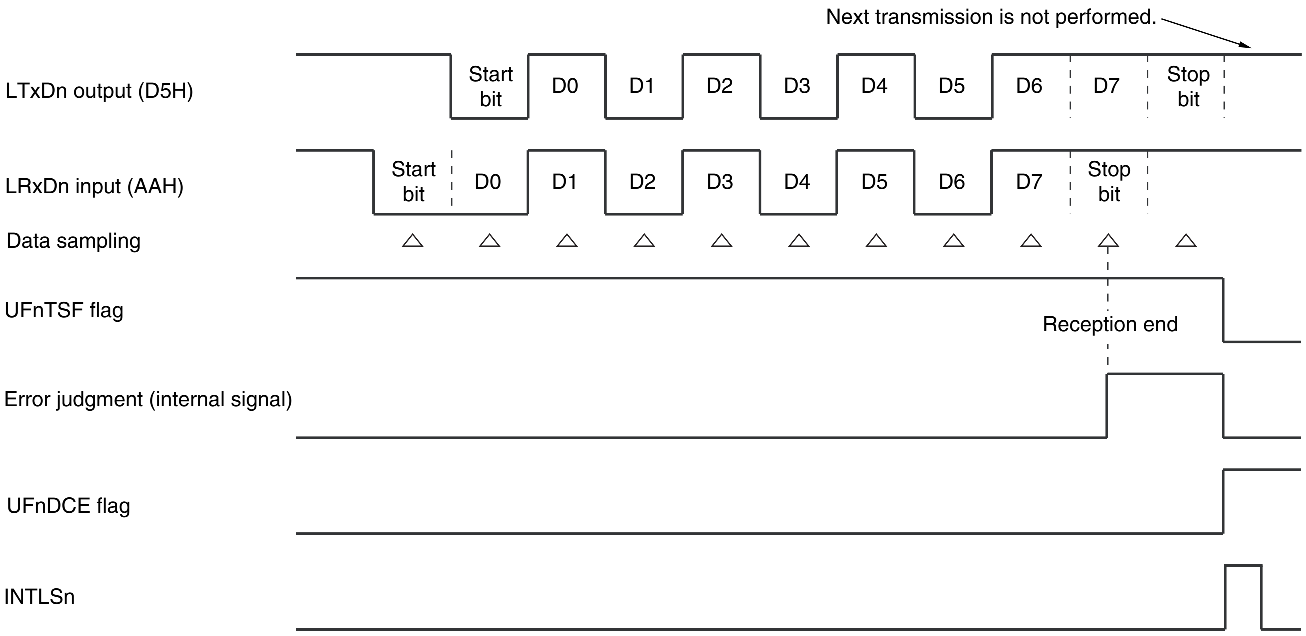


Remark n = 0, 1

(2) Early end of reception processing

When transmission is performed while reception is enabled (UFnTXE = UFnRXE = 1), a stop bit position detected in the reception processing, even though during transmission is judged as being abnormal and the UFnDCE bit is set (1) at the same time a status interrupt (INTLSn) is generated.

Figure 12-40. Timing Example of Consistency Error Occurrence due to Early End of Reception Processing



Remark n = 0, 1

12.5.9 BF reception mode select function

A mode for BF (break field) reception, which can be selected by using the LIN-UART operation mode selection bits (UFnMD1, UFnMD0), is provided.

(1) Normal UART mode (UFnMD1 and UFnMD0 = 00B)

In normal UART mode (UFnMD1 and UFnMD0 = 00B), a new BF is only recognized when the system is waiting for a BF to be successfully received (UFnBRF = 1).

<R> When BF reception is ended normally, the completion interrupt (INTLRn) of reception occurs.

If the system is not waiting for a BF to be successfully received (UFnBRF = 0), framing or overrun errors are detected at the data's stop bit position (bit 10) (see **Figure 12-41**). If an overrun error has not occurred, the received data is stored in the UFnRX register. If the system is waiting for a BF to be successfully received (UFnBRF = 1), framing or overrun errors are not detected and the received data is not stored in the UFnRX register. If UFnBRF = 0 and reception is stopped when data or the BF stop bit is transmitted, the data consistency error interrupt is issued and the flag is changed when transmission of the bit following the stop bit starts (see **12.5.8 (2)**). If reception is in progress when the stop bit is transmitted, the data consistency error interrupt is issued and the flag is changed when transmission of the stop bit starts (see **12.5.8 (1)**). On the other hand, if UFnBRF = 1 and reception is stopped when the stop bit is transmitted, the data consistency error interrupt is issued and the flag is changed when transmission of the bit following the stop bit starts (see **Figure 12-42**) and if reception is in progress when the stop bit is transmitted, the data consistency error interrupt is issued and the flag is changed when the rising edge of the input data following the stop bit is detected (see **Figure 12-43**).

Caution The successful BF reception flag (UFnBSF) is not set in normal UART mode.

Figure 12-41. Timing of Judging Framing or Overrun Error in Normal UART Mode

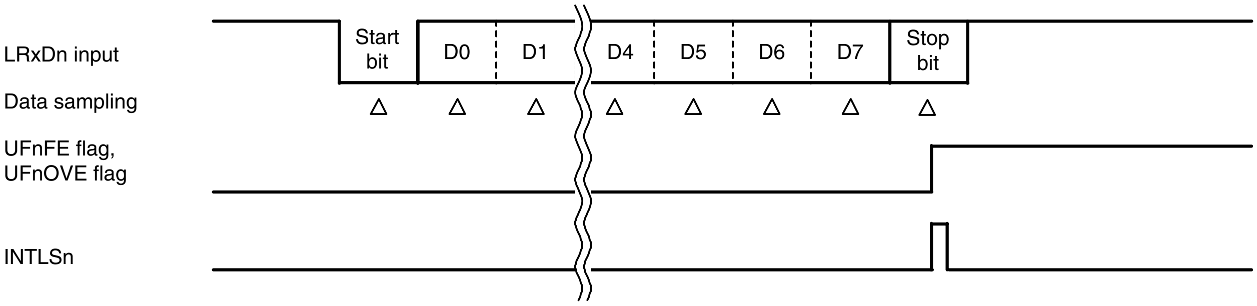


Figure 12-42. Timing of Occurrence of Data Consistency Error When BF Is Transmitted When UFnBRF = 1 (When Reception Is in Progress After Transmission of Stop Bit Has Stopped (Previous Input Data = 1))

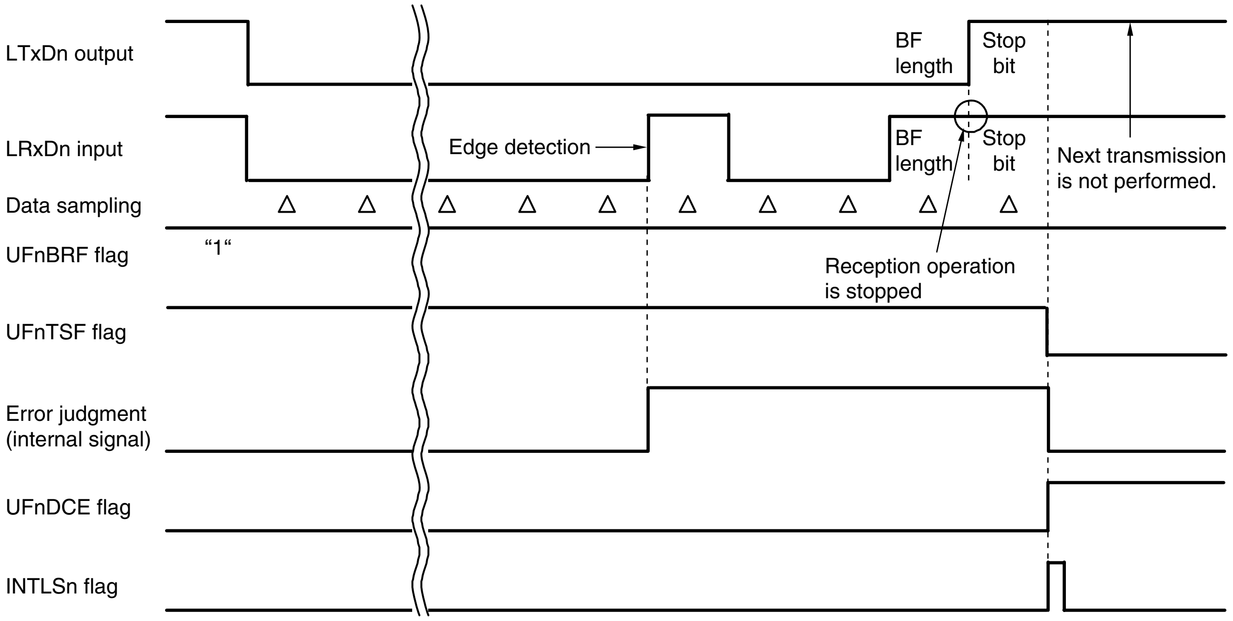
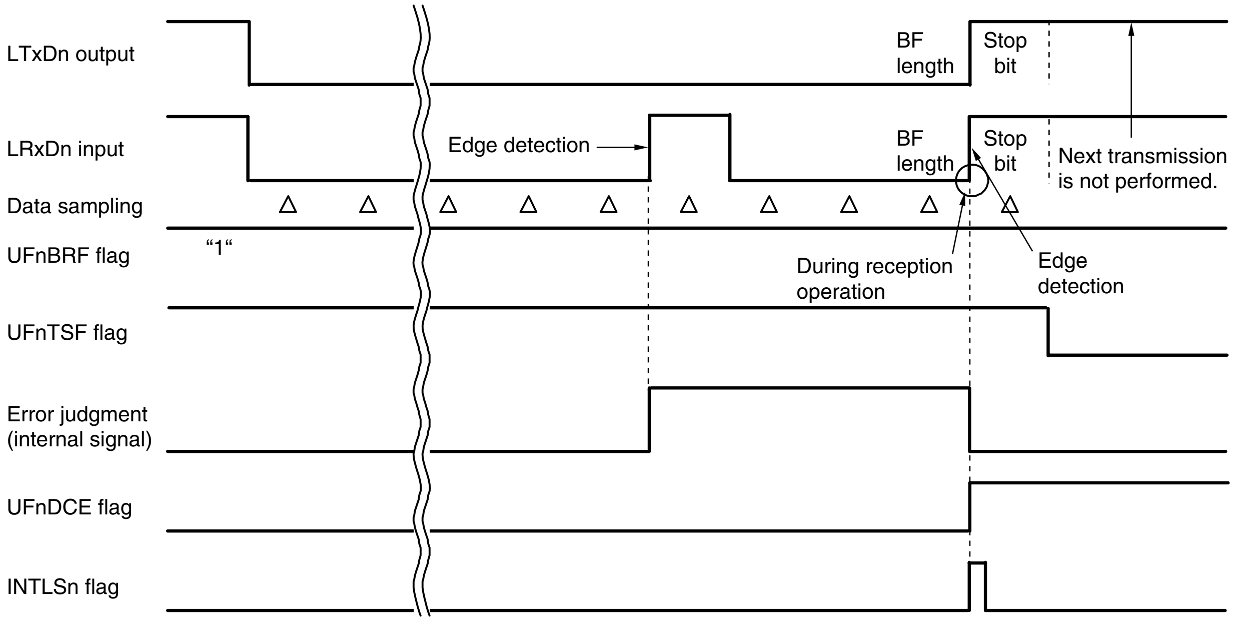


Figure 12-43. Timing of Occurrence of Data Consistency Error When BF Is Transmitted When UFnBRF = 1 (When Reception Is in Progress After Transmission of Stop Bit Has Started (Previous Input Data = 0))



(2) BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B)

If BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B) is set, a mode that recognizes a new BF is entered during data communication in addition to when waiting for successful BF reception (UFnBRF = 1). When not waiting for successful BF reception (UFnBRF = 0) and when a low level has been detected at the data stop bit position (10th bit), judging a framing error or an overrun error is being waited for until input data becomes high level, because a new BF may be undergoing reception. If the successive-low-level period is less than 11 bits, it is judged as error detection (see **Figure 12-44**). If not an overrun error, the first eight bits of receive data are stored into the UFnRX register. At this time, a successful BF reception flag (UFnBSF) is not set. When waiting for successful BF reception (UFnBRF = 1), detecting framing or overrun errors and storing receive data into the UFnRX register are not performed.

On the other hand, if the successive-low-level period is at least 11 bits, receiving of the new BF is judged successful and a successful BF reception flag (UFnBSF) is set (see **Figure 12-45**). Detection of framing or overrun errors is not performed. At this time, receive data is not stored into the UFnRX register.

If a reception operation is stopped when starting to transmit the stop bit of data or a BF while UFnBRF is “0”, the data consistency error interrupt and flag are changed when the bit following the stop bit is started (see **12.5.8 (2)**).

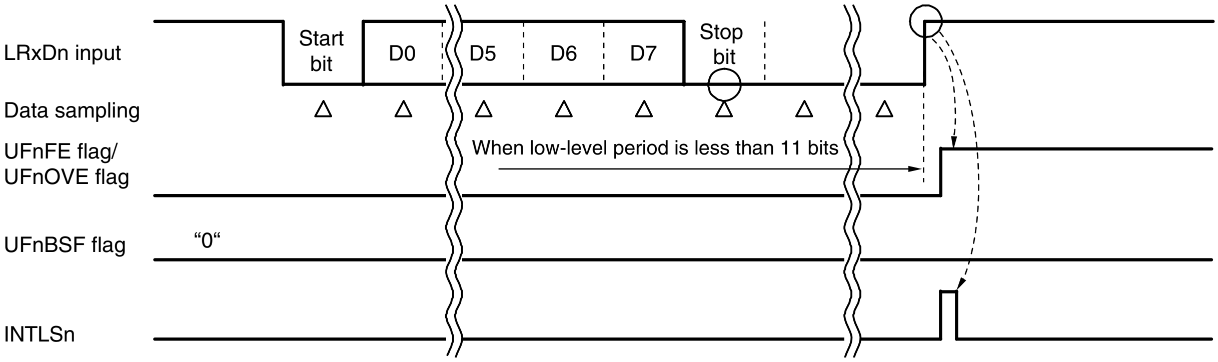
If a reception operation is being performed when starting to transmit the stop bit, it is performed when input data “1” is detected at a position following the stop bit (see **12.5.8 (1)** and **Figure 12-46**).

<R> On the other hand, when input data “1” is detected during BF transmission by UFnBRF = 1, it is carried out after the completion of transmitting of a stop bit of the 1st bit (see **Figure 12-47**).

In after the completion of BF transmitting, it is carried out in the timing of a bit which detected “1” (see **Figure 12-48**).

Caution To set to BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B), be sure to set the UFnDCS bit of the UFnOPT1 register also to “1”.

Figure 12-44. Framing Error/Overrun Error Judgment Timing upon BF Reception Failure (When UFnBRF = 0)



Remark n = 0, 1

Figure 12-45. Status Interrupt Occurrence Timing upon Successful BF Reception (When UFnBRF = 0)

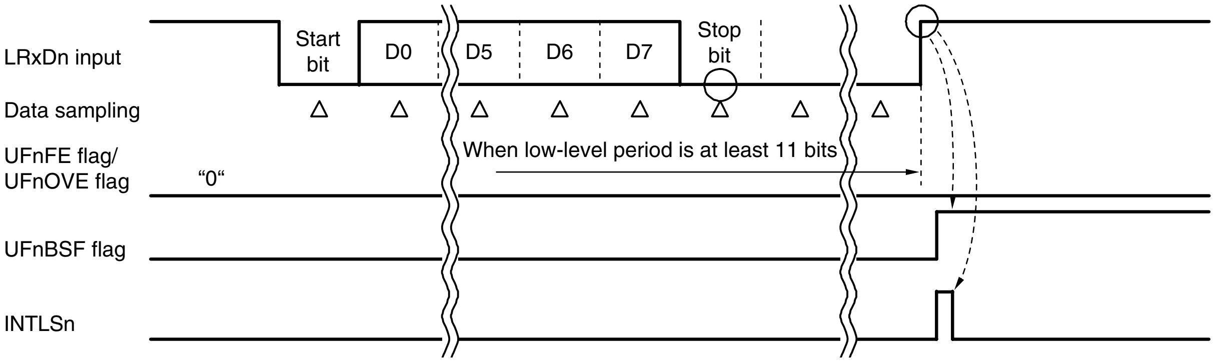
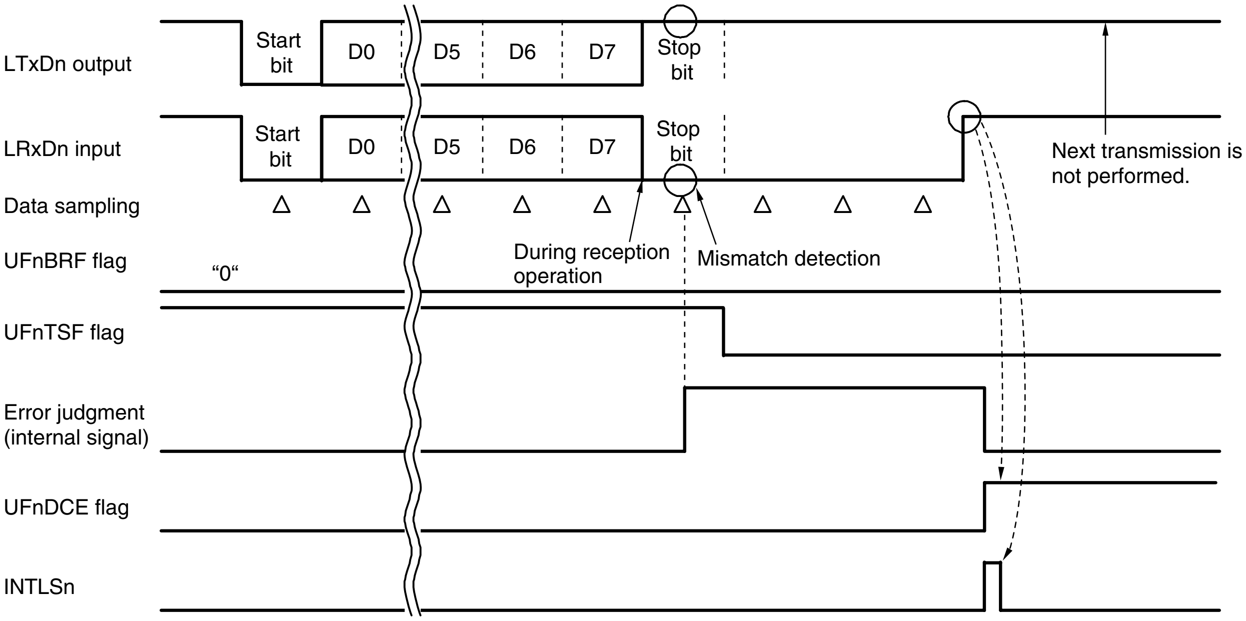
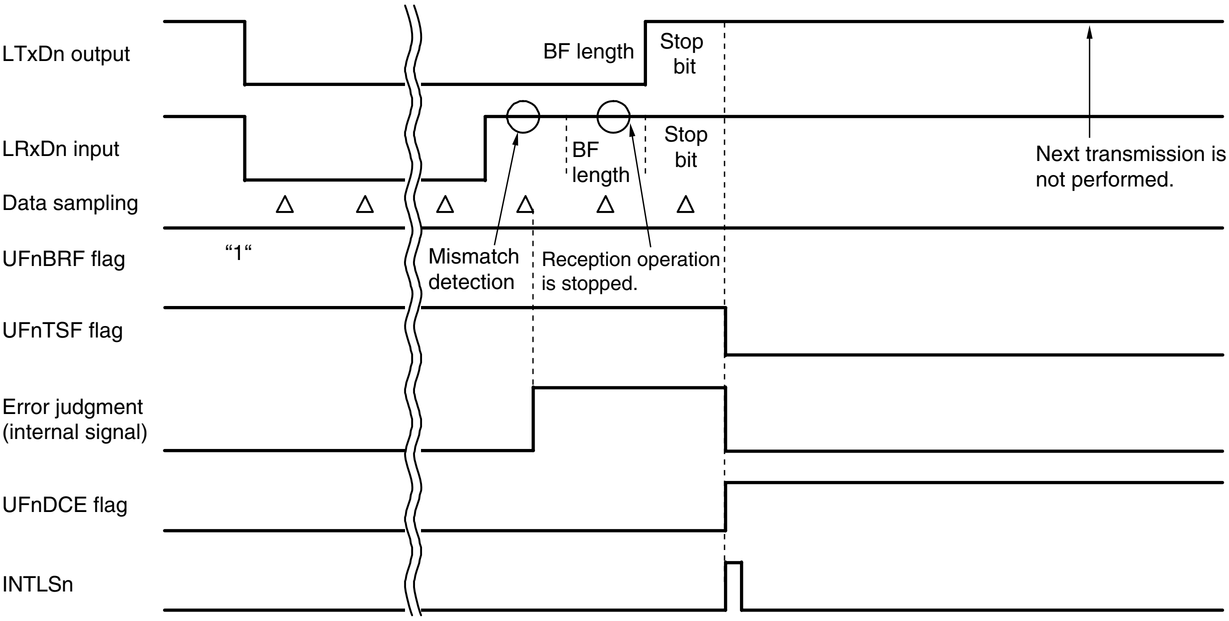


Figure 12-46. Example of Data Consistency Error Occurrence Timing When UFnBRF = 0

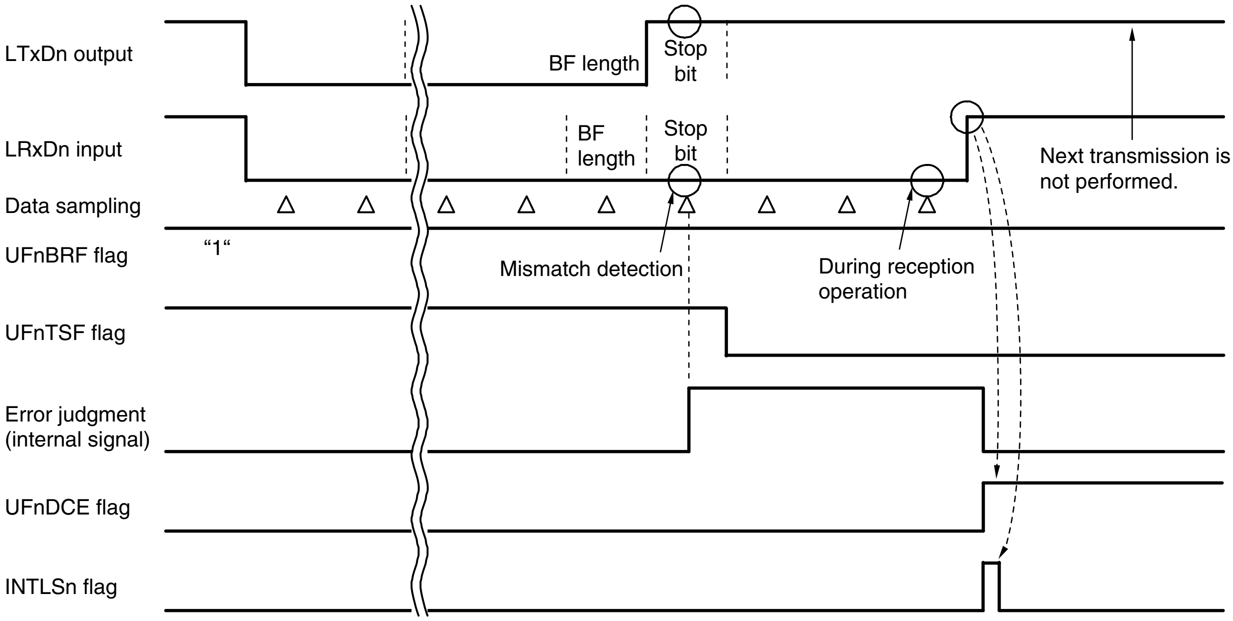


Remark n = 0, 1

<R> **Figure 12-47. Example of Consistency Error Occurrence Timing During BF Transmission When UFnBRF = 1 (When "1" is detected during BF transmission)**



<R> **Figure 12-48. Example of Consistency Error Occurrence Timing During BF Transmission When UFnBRF = 1 (If During Reception Operation When Input Data "1" Is Detected After Stop Bit (When "1" is detected during BF transmission)**



Remark n = 0, 1

12.5.10 Status interrupt generation sources

Status interrupt generation sources include parity errors, framing errors, overrun errors, data consistency errors which occur only during LIN communication, successful BF reception, ID parity errors, checksum errors, and response preparation errors which occur only in automatic baud rate mode, and ID matches and expansion bit detections which occur only when expansion bits are enabled. When these sources are detected, a status interrupt request signal (INTLSn) is generated. The type of a generation source can be referenced by using the status register (UFnSTR). The content of processing is determined by referencing the UFnSTR register in the status interrupt servicing routine.

Status flags must be cleared by writing “1” to the corresponding bits (excluding the UFnTSF and UFnRSF bits of the UFnSTC register) by using software.

The status interrupt generation timing and status flag change timing differ, depending on the mode setting and generation source.

Table 12-3. Status Interrupt Generation Sources

Status Flag	Generation Source	Description
UFnPE	Parity error	The parity calculation result of receive data and the value of the received parity bit do not match.
UFnFE	Framing error	No stop bit is detected. (A low level is detected at a stop bit position.)
UFnOVE	Overrun error	The next data reception is completed before the receive data transferred to the receive data register is read.
UFnDCE	Data consistency error	The data consistency check selection bit (UFnDCS) is set, and the values of transmit data and receive data do not match during data transmission. Transmission operation and reception operation are out of synchronization.
UFnBSF	Successful BF reception	A new BF is successfully received when in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B). (This occurs also when the master transmits a BF.)
UFnIPE	ID parity error	Either parity bit of the received PID includes an error.
UFnCSE	Checksum error	The result of comparing the checksum received during response reception and the automatically calculated result is illegal.
UFnRPE	Response preparation error	Response preparation could not be performed before reception of the first byte by a response was completed.
UFnIDM	ID match	When the following conditions are satisfied: - Comparison of expansion bit data is enabled (UFnEBC = 1). - The expansion bit is at the level set by using the expansion bit detection level selection bit (UFnEBL). - The received data matches the value of the UFnID register.
UFnEBD	Expansion bit detection	The level set by using the expansion bit detection level select bit (UFnEBL) is detected at a receive data expansion bit.

Remark n = 0, 1

The following processing is required depending on the generation source when a status interrupt is generated.

- Parity error, data consistency error
  False data has been received, so read the received data and then discard it. Then perform communication again. If the received data is not read, an overrun error might occur when reception ends next time. For a data consistency error, a data conflict may also be possible.

- Framing error
  The stop bit could not be detected normally, or a bit offset may have occurred due to false detection of the start bit. Furthermore, the baud rate may be offset from that of the transmission side or a BF of insufficient length may have been received in LIN communication.
  When framing errors occur frequently, a bit or the baud rate may be offset, so perform initialize processing on both the transmission side and reception side, and restart communication. Furthermore, to receive the next data after a framing error has occurred, the reception pin must become high level once.

- Overrun error
  Data of one frame that was received immediately before is discarded, because the next reception is completed before receive data is read. Consequently, the data must be retransmitted.

- Successful BF reception
  Preparation for starting a new frame slot must be performed, because a new BF has been received successfully.

- ID parity error
  Set a request bit without a response (UFnNO), because the received PID is illegal. Afterward, do not perform response transmission or reception, wait for the next BF to be received, and ignore that frame.

- Checksum error
  Discard the received response (data field), because it is illegal.

- Response preparation error
  Wait for the next BF to be received and ignore that frame, because response processing cannot be performed normally.

- ID match
  Receive data of the expansion bit of a level set by using the UFnEBL bit has matched with the UFnID register setting value. Perform, therefore, corresponding processing such as disabling expansion bit data comparison (UFnEBC = 0) to receive subsequent data.

- Expansion bit detection
  Perform corresponding processing such as preparing for starting DMA transfer, because receive data of the expansion bit of a level set by using the UFnEBL bit has been received.

Caution Status flags are an accumulation of all sources that have been generated after the status flag has been cleared, and do not reflect the latest state. Consequently, the above-mentioned processing must be completed before the next reception is completed and the status flag must be cleared.

The following table shows examples of processing corresponding to statuses when performing LIN communication.

Table 12-4. Examples of Processing Corresponding to Statuses During LIN Communication (When in BF Reception Enable Mode During Communication (UFnMD1, UFnMD0 = 10B) and When UFnDCS = 1)

UFnBSF	UFnDCE	UFnFE	UFnOVE	Status	Processing Example
1	1	×	×	A mismatch is detected between transmit and receive data during BF transmission in master operation. Successive low levels of at least 11 bits are received. The transmission is not performed even if the next data transmission has been prepared.	• The next data (Sync field) transmission is not performed and waiting for the next time schedule is performed, because the other party of communication may not have been able to recognize the BF. • The other party of communication may not have been able to recognize the BF, but all status flags are cleared and the next data is written to transmit the next data (Sync field).
1	0	×	×	BF transmission and BF reception are performed successfully in master operation.	Processing to transmit the next data (Sync field) is performed.
				BF reception is performed successfully in slave operation.	Processing to receive the next data (Sync field) is performed.
0	1	×	×	BF transmission or data (including an SF or a PID) transmission has failed in master operation. Even if transmission of the next data or BF has been prepared, the transmission will not be performed.	Subsequent transmit and receive data is discarded, all status registers are cleared, and the system waits for the next time schedule.
				Data transmission has failed in slave operation. Even if transmission of the next data has been prepared, the transmission will not be performed.	Subsequent transmit and receive data is discarded, all status registers are cleared, and the system waits for the next time schedule.
0	0	1	×	A framing error has been detected during data reception.	Processing when a framing error has been detected is performed.
0	0	×	1	An overrun error has been detected during data reception. The single data that was received immediately before has been discarded.	Processing when an overrun error has been detected is performed.

Cautions 1. Clear all status flags that have been set for any processing.

2. When an error is detected in LIN communication (including when BF reception has been performed successfully when BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B) has been set), a status interrupt request signal (INTLSn) is generated instead of a reception complete interrupt request signal (INTLRn) and a status flag is set according to the communication status.

Remarks 1. ×: don’t care

2. n = 0, 1

12.5.11 Transmission start wait function

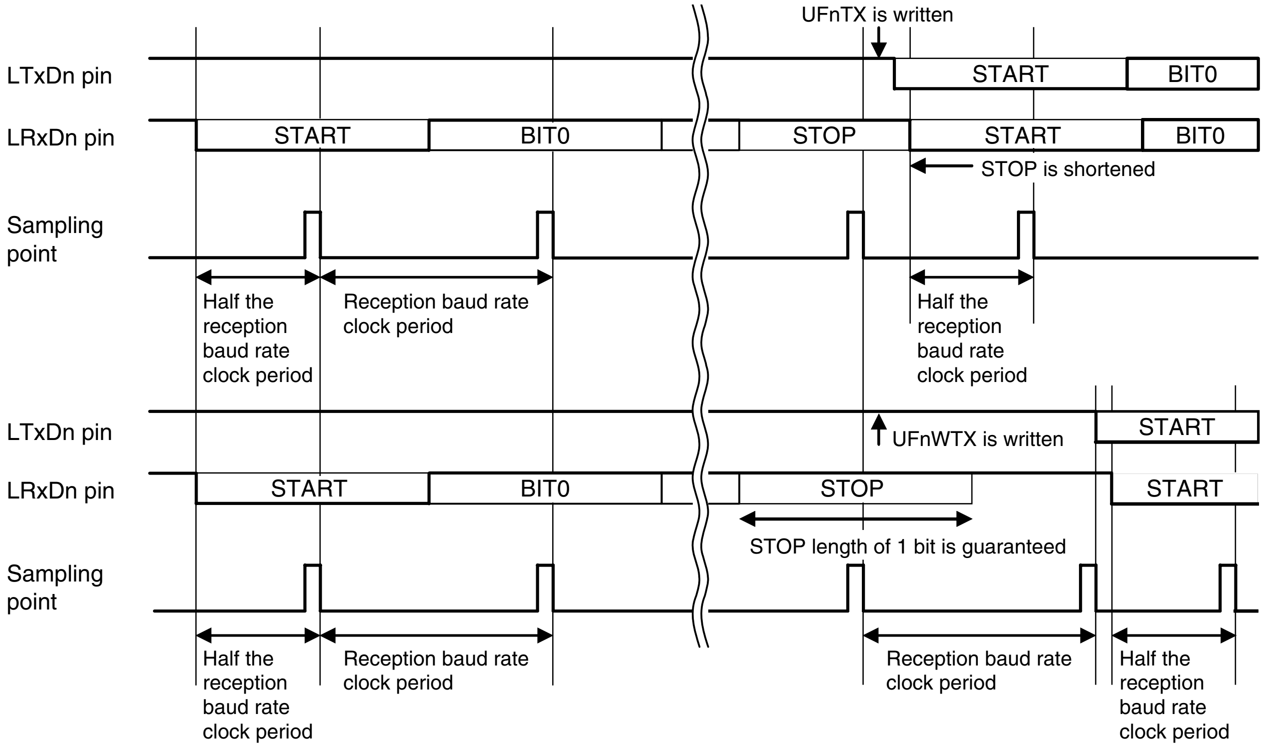
The 78K0R/Hx3 is provided with a function to guarantee the stop bit length of reception when reception is switched to transmission to perform LIN communication.

To delay starting of transmission until completion of the stop bit of reception, write data to the UFnWTX register which is a wait-dedicated register, instead of writing transmit data to the UFnTX register as a transmission start request.

In this case, starting transmission is being waited for one bit until the stop bit of receive data has ended for sure.

Note that only a wait of one bit is performed, even if the stop bit length has been set to two bits by using the stop bit length select bit (UFnSL).

Figure 12-49. When Transmit Data Has Been Written During Stop Bit of Receive Data


UFnTX is written
LTxDn pin
START
BIT0
LRxDn pin
START
BIT0
STOP
START
BIT0
STOP is shortened
Sampling point
Half the reception baud rate clock period
Reception baud rate clock period
Half the reception baud rate clock period
LTxDn pin
UFnWTX is written
START
LRxDn pin
START
BIT0
STOP
START
STOP length of 1 bit is guaranteed
Sampling point
Half the reception baud rate clock period
Reception baud rate clock period
Reception baud rate clock period
Half the reception baud rate clock period

Cautions 1. When LIN communication is not performed, accessing the UFnWTX register is prohibited.

2. Writing to the UFnWTX register is prohibited except when reception is switched to transmission (such as during transmission).

Remark n = 0, 1

12.6 UART Buffer Mode

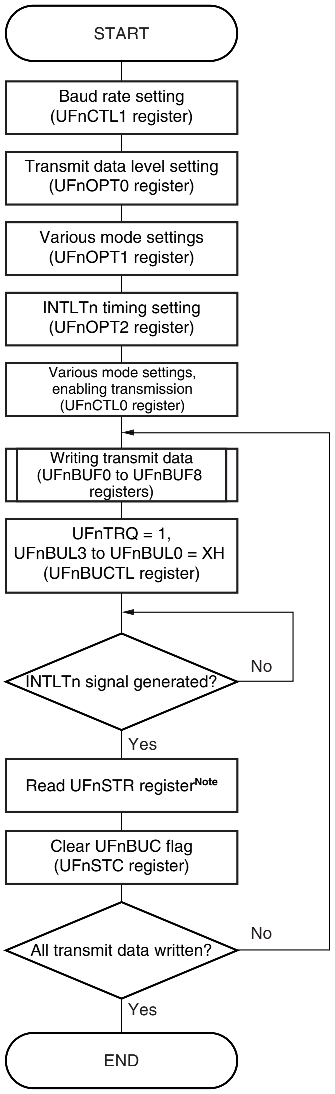
The 78K0R/Hx3 is provided with a 9-byte transmission buffer that can be used for normal UART communication (UFnMD1, UFnMD0 = 00B).

Remark n = 0, 1

12.6.1 UART buffer mode transmission

The following figure shows the procedure for transmitting data in UART buffer mode.

Figure 12-50. UART Buffer Mode Transmission Processing Flow



Note This can be omitted.

Cautions 1. Set the following values when performing BF transmission.

Expansion bits are disabled (UFnEBE = 0).
Normal UART mode (UFnMD1, UFnMD0 = 00B).
Data consistency checking is disabled (UFnDCS = 0).
Waiting for buffer transmission start is disabled (UFnTW = 0).
Continuation of transfer is disabled (UFnCON = 0).
Request bits without responses are present (UFnNO = 0).
Reception requests are disabled (UFnRRQ = 0).

2. When clearing the UFnBUC flag, be careful not to clear other reception flags.

Remarks 1. See **(2)** of **12.11 Cautions on Use for details of starting LIN-UART**.

2. X: don't care, n = 0, 1

When transferring the number of bytes (1 to 9) set to the buffer length bit (UFnBUL3 to UFnBUL0) has ended, a transmission interrupt request signal (INTLTn) is output. When the buffer length bit is set to "0" or "10 to 15", transfer of nine bytes is performed.

Writing data to the transmit data register (UFnTX) during transmission in buffer mode is prohibited.

To stop transfer midway, write "0" to the transmission enable bit (UFnTXE). Data transmission processing is stopped and the UFnTRQ bit and UFnTSF flag are cleared.

Figure 12-51. UART Buffer Mode Transmission Example (UFnITS = 0)

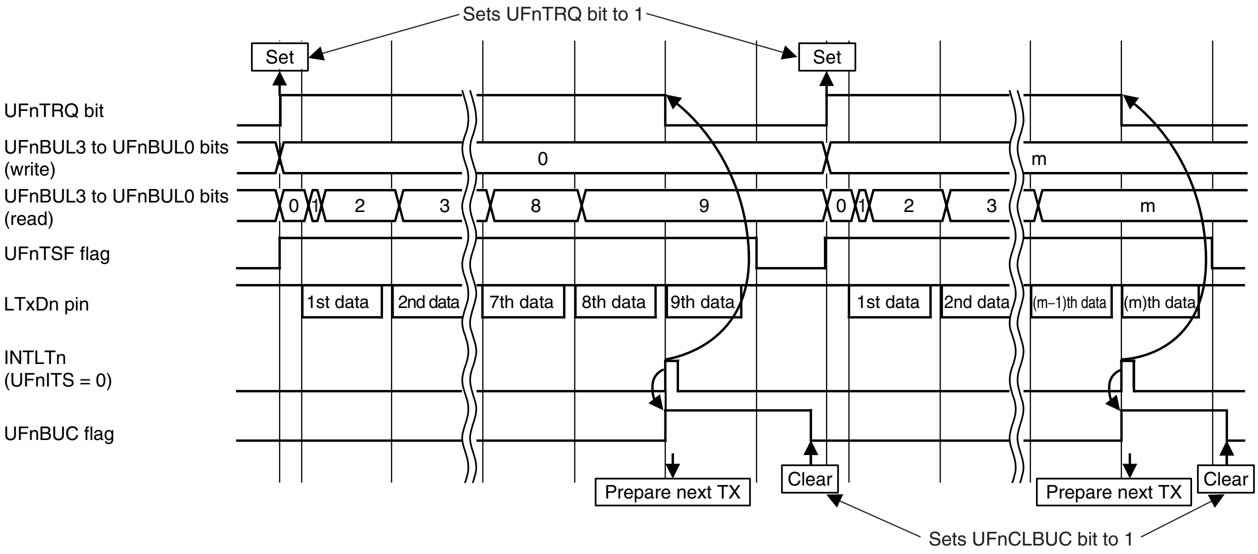
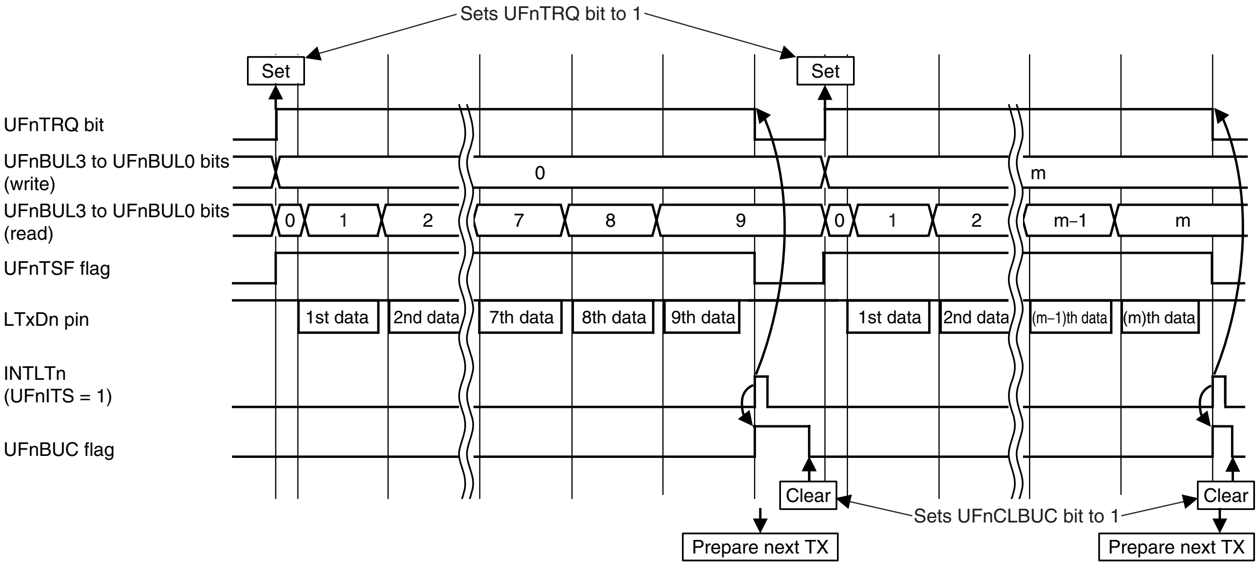


Figure 12-52. UART Buffer Mode Transmission Example (UFnITS = 1)



Remark n = 0, 1, m = 1 to 9

12.7 LIN Communication Automatic Baud Rate Mode

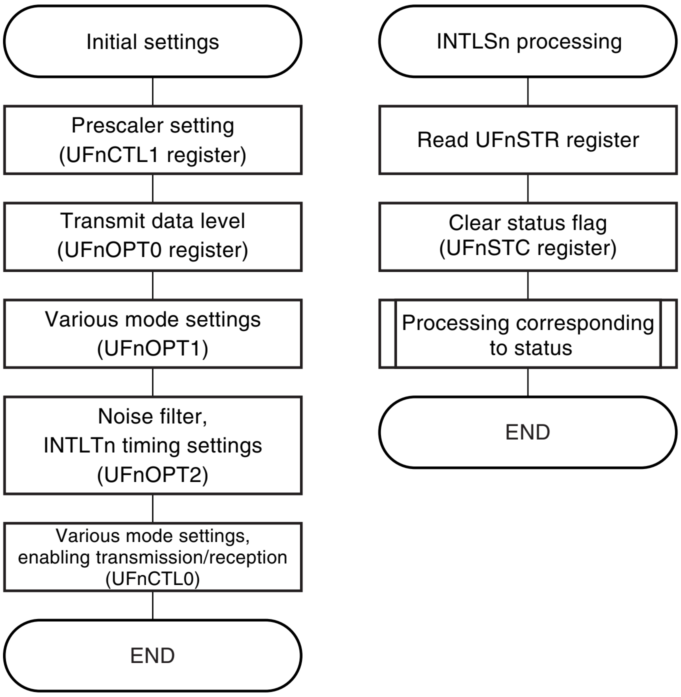
In LIN communication automatic baud rate mode, a BF and an SF are automatically detected and the baud rate is set according to the measurement result of the SF.

When UFnMD1 and UFnMD0 are set to “11B”, operation is performed in automatic baud rate mode.

Operation can be performed with the baud rate at 2,400 bps to 128 kbps. Set to 8 to 12 MHz the clock (prescaler clock) that has been divided by using a prescaler. At that time, the setting values of UFnPRS2 to UFnPRS0 must be calculated from the f_{CLK} frequency and initial settings must be performed.

When using LIN-UART as the master, using automatic baud rate mode (UFnMD1, UFnMD0 = 11B) is prohibited.

Figure 12-53. Basic Processing Flow Example of LIN Communication Automatic Baud Rate Mode (1/2)


Initial settings
Prescaler setting (UFnCTL1 register)
Transmit data level (UFnOPT0 register)
Various mode settings (UFnOPT1)
Noise filter, INTLTn timing settings (UFnOPT2)
Various mode settings, enabling transmission/reception (UFnCTL0)
END
INTLSn processing
Read UFnSTR register
Clear status flag (UFnSTC register)
Processing corresponding to status
END

Cautions 1. Set the following values when performing LIN communication automatic baud rate mode.

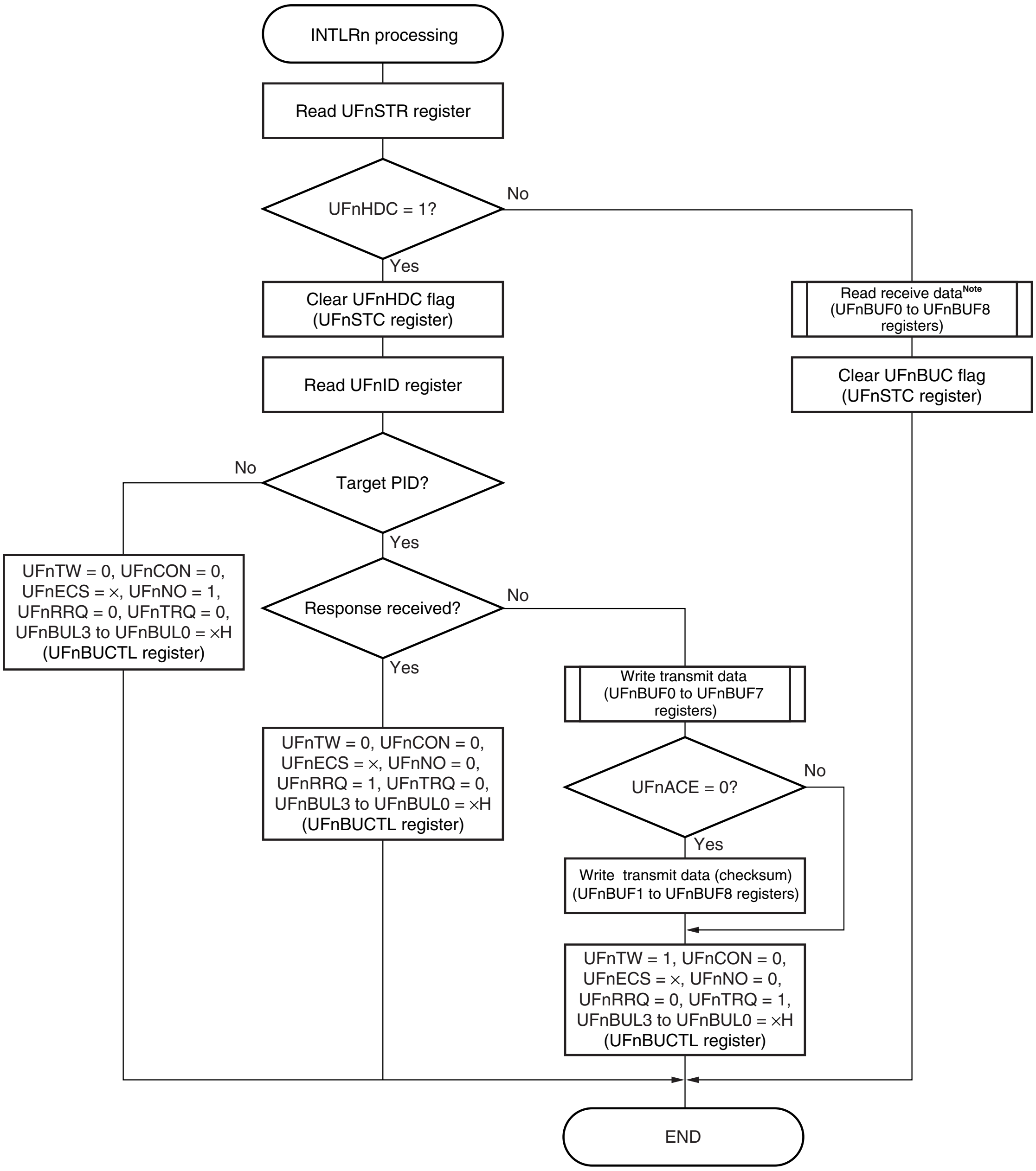
The transmit and receive data levels are normal input (UFnTDL = UFnRDL = 0).
Expansion bits are disabled (UFnEBE = 0).
Automatic baud rate mode (UFnMD1, UFnMD0 = 11B) as the mode.
Consistency check selection (UFnDCS = 1).
Transmission interrupt is transmission start (UFnITS = 0).
Communication direction control is LSB first (UFnDIR = 1).
The parity selection bit is received without parity (UFnPS1, UFnPS0 = 00B).
The data character length is 8 bits (UFnCL = 1).
Transmit data register is default value (UFnTX = 0000H).

2. Set the UFnPRS2 to UFnPRS0 bits so that the clock that has been divided by using a prescaler is 8 to 12 MHz.

Remarks 1. See **(2)** of **12.11 Cautions on Use for details of starting LIN-UART**.

2. n = 0, 1

Figure 12-53. Basic Processing Flow Example of LIN Communication Automatic Baud Rate Mode (2/2)


INTLRn processing
Read UFnSTR register
UFnHDC = 1?
No
Yes
Clear UFnHDC flag (UFnSTC register)
Read UFnID register
Target PID?
No
Yes
UFnTW = 0, UFnCON = 0, UFnECS = ×, UFnNO = 1, UFnRRQ = 0, UFnTRQ = 0, UFnBUL3 to UFnBUL0 = ×H (UFnBUCTL register)
Response received?
No
Yes
UFnTW = 0, UFnCON = 0, UFnECS = ×, UFnNO = 0, UFnRRQ = 1, UFnTRQ = 0, UFnBUL3 to UFnBUL0 = ×H (UFnBUCTL register)
Write transmit data (UFnBUF0 to UFnBUF7 registers)
UFnACE = 0?
No
Yes
Write transmit data (checksum) (UFnBUF1 to UFnBUF8 registers)
UFnTW = 1, UFnCON = 0, UFnECS = ×, UFnNO = 0, UFnRRQ = 0, UFnTRQ = 1, UFnBUL3 to UFnBUL0 = ×H (UFnBUCTL register)
Read receive dataNote (UFnBUF0 to UFnBUF8 registers)
Clear UFnBUC flag (UFnSTC register)
END

Note This can be omitted.

Cautions 1. When the buffer length bits (UFnBUL3 to UFnBUL0) have been set to “0” or “10 to 15”, reception or transmission of nine bytes is performed. When the buffer length is set to “1 to 8”, buffers of the number of bytes set are used in ascending order of the buffer numbers.

Example: When UFnBUL3 to UFnBUL0 are set to “1”, data is always stored only into the UFnBUF0 register.

2. Do not set the UFnRRQ bit until reading of received data is completed, because, when the UFnRRQ bit is set, storing (overwriting) into a buffer is performed even if reading receive data has not ended. <R>

3. Setting (1) the UFnTW bit is prohibited, except when operation is switched to response transmission after header reception.

Remark ×: don’t care, n = 0, 1

If a PID stored into the UFnID register is not a target when header reception is completed (UFnHDC = 1), the UFnNO bit is set and subsequent transmission and reception processing are stopped (responses are ignored).

<R> For a response reception PID, the UFnRRQ bit is set at the same time as the response data length (UFnBUL3 to UFnBUL0), and response reception processing is performed.

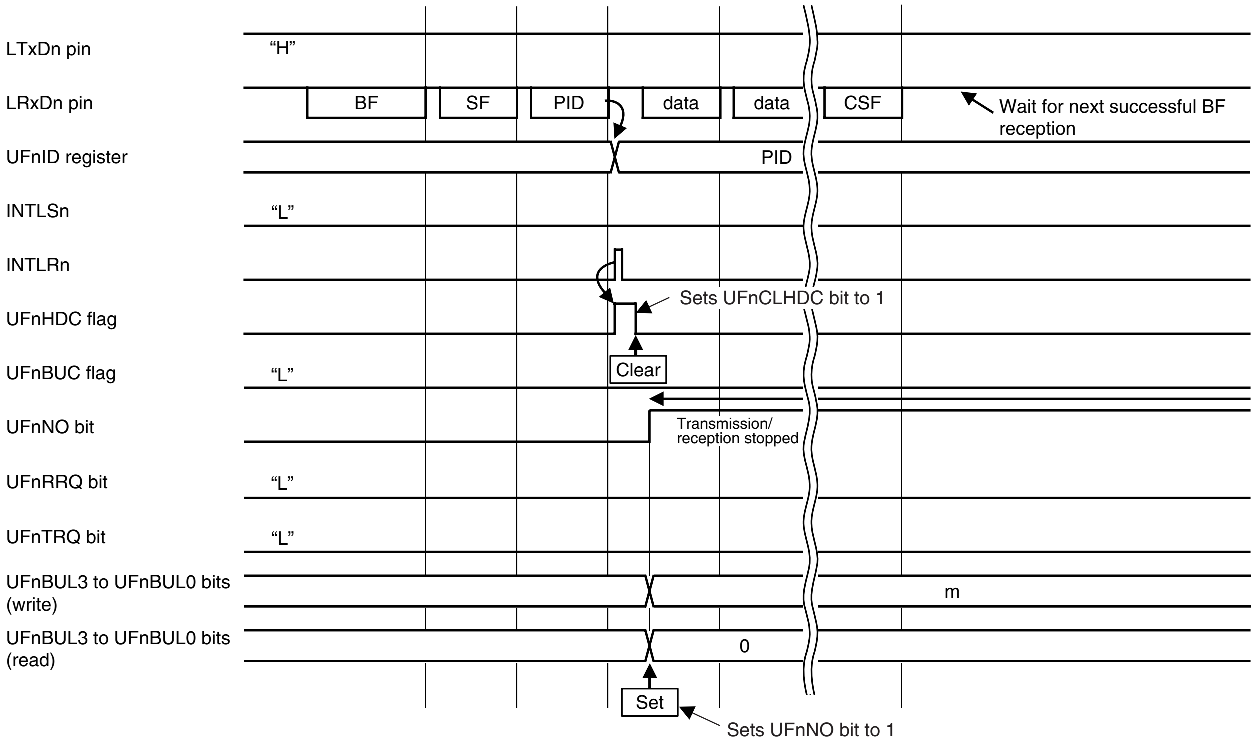
<R> For a response transmission PID, the UFnTRQ bit is set at the same time as the response data length (UFnBUL3 to <R> UFnBUL0), and response transmission processing is performed, after transmit data has been set to a buffer. At that time, the receive data will be stored in the UFnRX register. However, no overrun error will occur even if the receive data is not read.

Perform processing (setting the UFnNO, UFnRRQ, or UFnTRQ bit) for the PID before receiving the first byte of the response is completed. Otherwise, a response preparation error occurs. See **12.7.2 Response preparation error detection function** for details.

During response reception and response transmission also, when a status interrupt request signal (INTLSn) has been generated due to an error, transmission and reception operations are stopped and waiting for the next BF reception is performed.

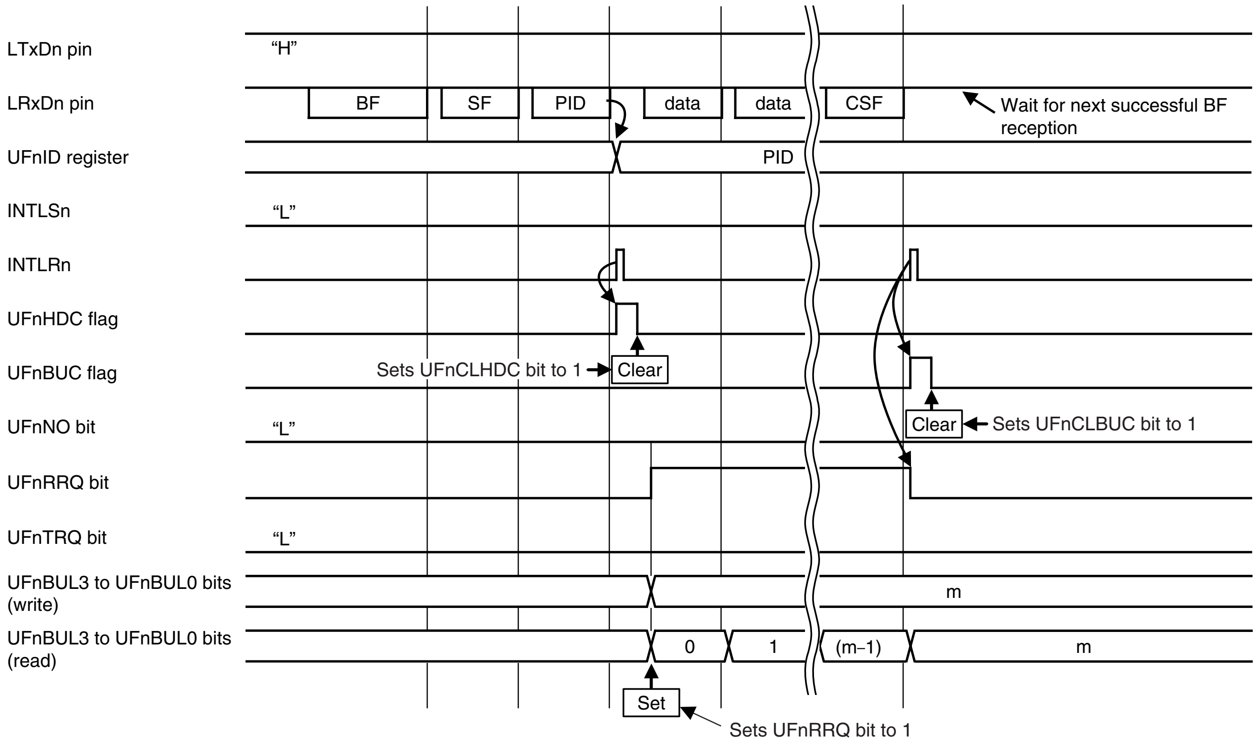
In automatic baud rate mode, no overrun error occurs, because a buffer is used (the UFnRX register is not used).

Figure 12-54. LIN Communication Automatic Baud Rate Mode (Non-Target PID)



Remark n = 0, 1, m = 1 to 9

Figure 12-55. LIN Communication Automatic Baud Rate Mode (Response Reception)



An example of how reception results are stored into a buffer when 8-byte data is received (UFnBUL3 to UFnBUL0 = 9) and when 3-byte data is received (UFnBUL3 to UFnBUL0 = 4) are shown below.

(1) When 8-byte data is received
(UFnBUL3 to UFnBUL0 = 9)

	Reception results
UFnBUF8	Checksum
UFnBUF7	Data7
UFnBUF6	Data6
UFnBUF5	Data5
UFnBUF4	Data4
UFnBUF3	Data3
UFnBUF2	Data2
UFnBUF1	Data1
UFnBUF0	Data0

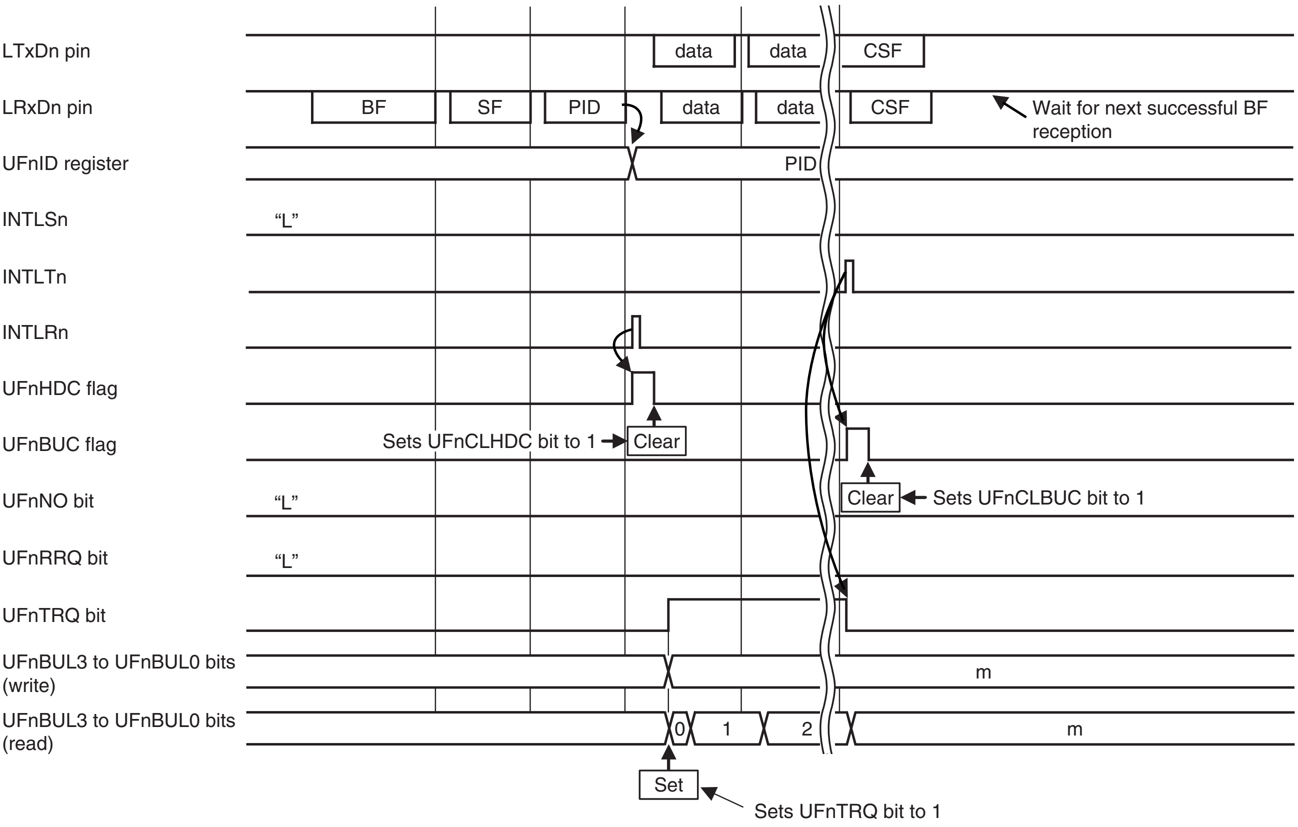
(2) When 3-byte data is received
(UFnBUL3 to UFnBUL0 = 4)

	Reception results
UFnBUF8	–
UFnBUF7	–
UFnBUF6	–
UFnBUF5	–
UFnBUF4	–
UFnBUF3	Checksum
UFnBUF2	Data2
UFnBUF1	Data1
UFnBUF0	Data0

Caution When UARTF is being used with the auto checksum feature enabled (UFnACE = 1), the checksum data is not stored in a buffer.

Remark n = 0, 1

<R> **Figure 12-56. LIN Communication Automatic Baud Rate Mode (Response Transmission)**



Examples of the buffer settings and the status of the buffer after 8 bytes of data have been transmitted (UFnBUL3 to UFnBUL0 = 9) and after 3 bytes of data have been transmitted (UFnBUL3 to UFnBUL0 = 4) are shown below.

(1) When 8-byte data is transmitted (UFnBUL3 to UFnBUL0 = 9)

	Buffer setting	Buffer status
UFnBUF8	TX Checksum	RX Checksum
UFnBUF7	Data7	Data7
UFnBUF6	Data6	Data6
UFnBUF5	Data5	Data5
UFnBUF4	Data4	Data4
UFnBUF3	Data3	Data3
UFnBUF2	Data2	Data2
UFnBUF1	Data1	Data1
UFnBUF0	Data0	Data0

(2) When 3-byte data is received (UFnBUL3 to UFnBUL0 = 4)

	Buffer setting	Buffer status
UFnBUF8	–	–
UFnBUF7	–	–
UFnBUF6	–	–
UFnBUF5	–	–
UFnBUF4	–	–
UFnBUF3	Checksum	Checksum
UFnBUF2	Data2	Data2
UFnBUF1	Data1	Data1
UFnBUF0	Data0	Data0

Caution **To enable the automatic checksum function (UFnACE = 1), checksum is not required to be set to the buffer by using software.**

Remark n = 0, 1

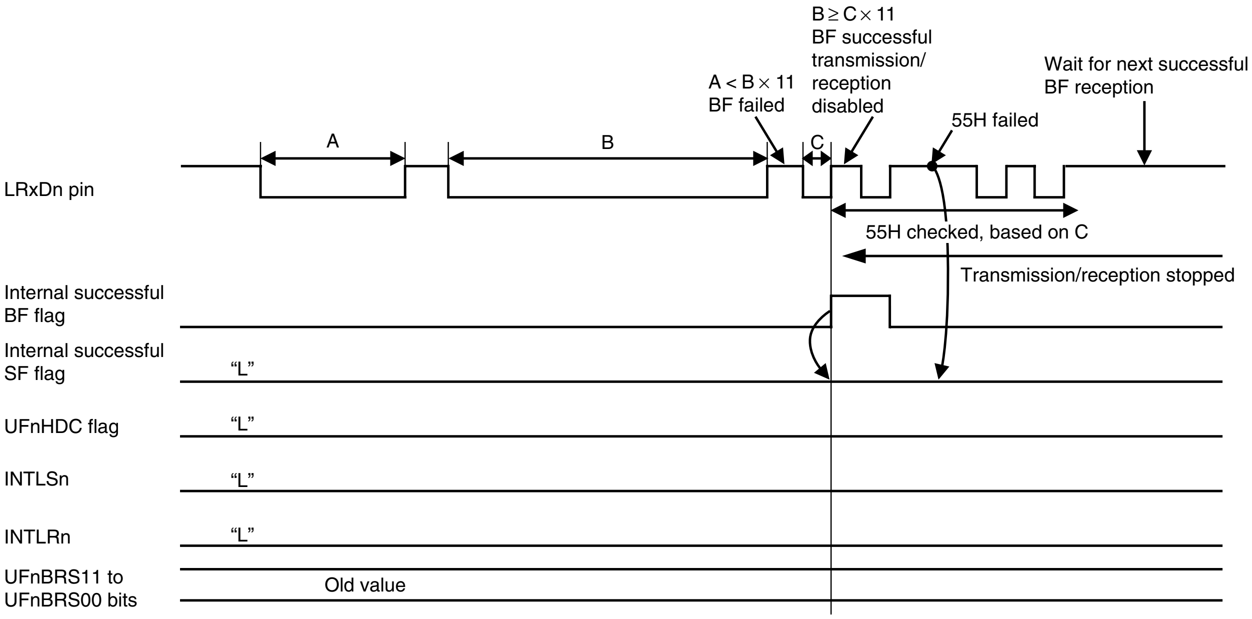
<R> 12.7.1 Automatic baud rate setting function

Received low-level widths are always measured when in automatic baud rate mode. BF detection is judged as being performed successfully when the first low-level width is at least 11 times the second low-level width, and it is checked that the data is 55H. If the data is confirmed to be 55H and the SF is judged to have been successfully received, reception is paused, the UFnBRS11 to UFnBRS00 bits are set again, and reception resumes after the start bit is detected.

When it has been confirmed that the data is 55H, successful SF detection is judged and baud rate setting results are automatically set to the UFnBRS11 to UFnBRS00 bits. At that time, the settings of the UFnPRS2 to UFnPRS0 bits are not changed. Afterward, the next data (PID) is received after transmission or reception processing has been enabled. A reception complete interrupt request signal (INTLRn) is generated when there are no errors upon PID reception completion (stop bit position), and an error flag is set and a status interrupt request signal (INTLSn) is generated when there is an error. A header reception completion flag (UFnHDC) is set when there are no errors upon PID reception completion (stop bit position), and a header reception completion flag (UFnHDC) is not set when there is an error. On the other hand, when the data is not 55H, SF detection is judged to have failed, the next BF (low level) reception is being waited for with the transmission or reception processing being stopped, and baud rate setting is not performed.

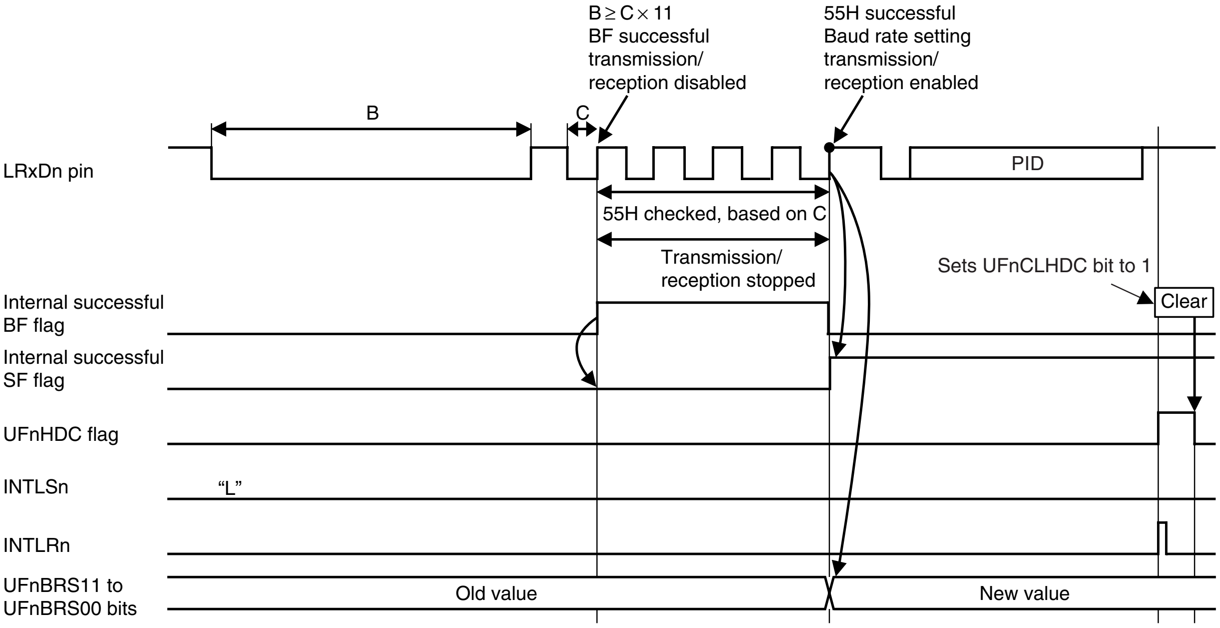
When the stop bit position of reception processing is reached while transmission or reception processing is enabled, errors such as framing errors and consistency errors are detected and a status interrupt request signal (INTLSn) may be generated. This is also applicable when a BF has been received during communication.

Figure 12-57. Example of BF/SF Reception Failure



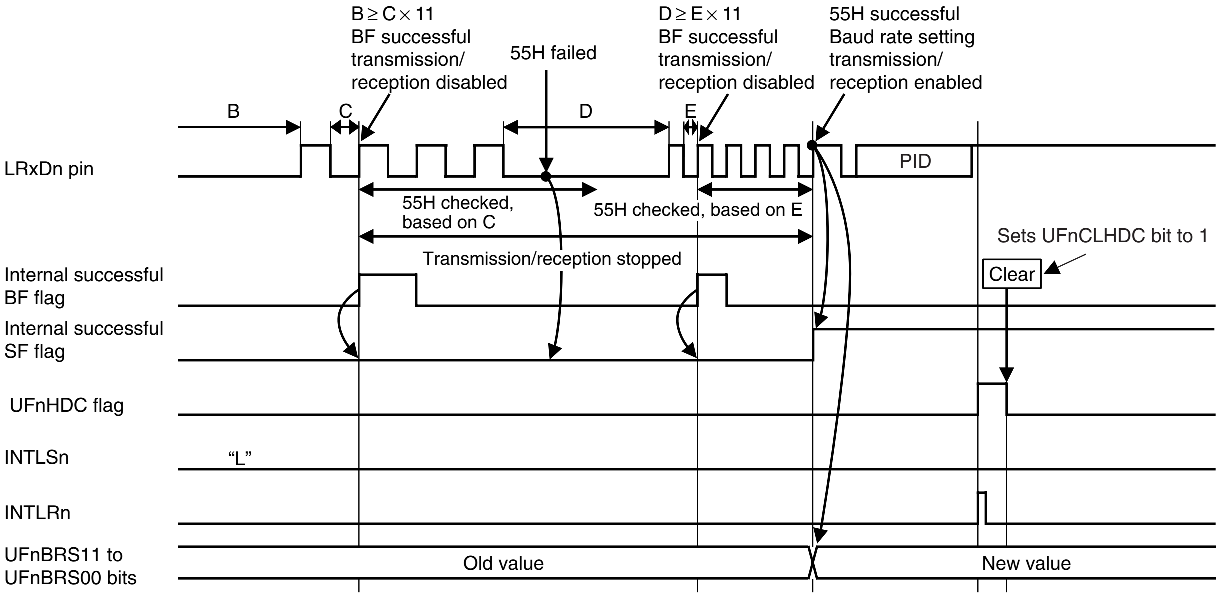
Remark n = 0, 1

Figure 12-58. Example of Successful BF, SF, and PID Reception



Caution When a PID reception error has occurred, a status interrupt request signal (INTLSn) is generated instead of a reception complete interrupt request signal (INTLRn) and other error flags (such as UFnFE and UFnIPE) change.

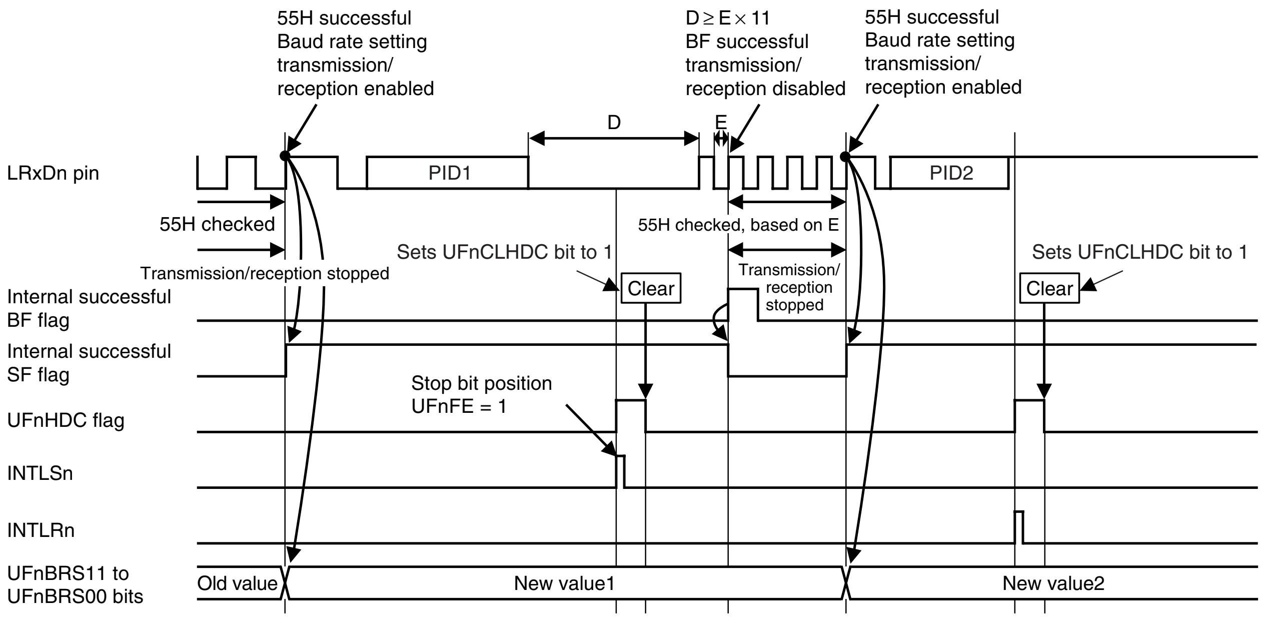
Figure 12-59. Example of Successful BF Reception During SF Reception (No PID Reception Error)



Caution When a PID reception error has occurred, a status interrupt request signal (INTLSn) is generated instead of a reception complete interrupt request signal (INTLRn) and other error flags (such as UFnFE and UFnIPE) change.

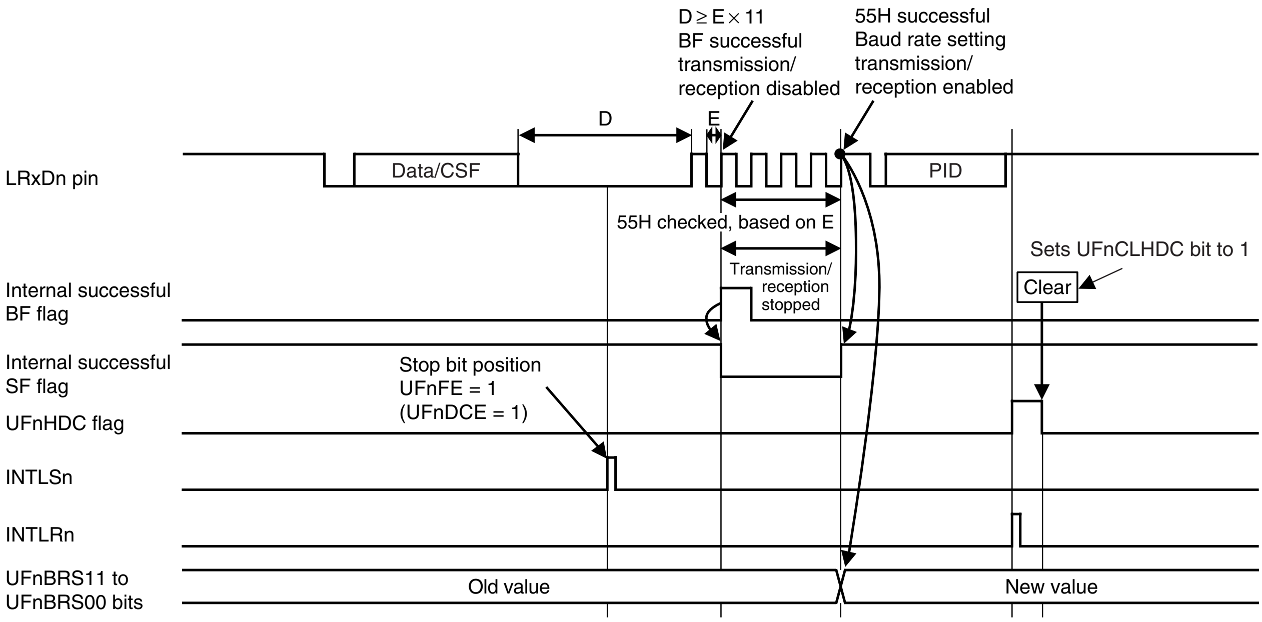
Remark n = 0, 1

Figure 12-60. Example of Successful BF Reception During PID Reception (No PID2 Reception Error)



Caution **If the PID1 stop bit position comes after the point where the internal successful BF flag has been set, the UFnHDC flag and error flags (such as UFnFE and UFnIPE) are not set, and INTLSn is also not generated.**

Figure 12-61. Example of Successful BF Reception During Data/CSF Reception (No PID Reception Error)



Caution **If the Data/CSF stop bit position comes after the point where the internal successful BF flag has been set, the UFnBUC flag and error flags (such as UFnFE, UFnDCE, UFnCSE, and UFnRPE) are not set, and INTLSn is also not generated.**

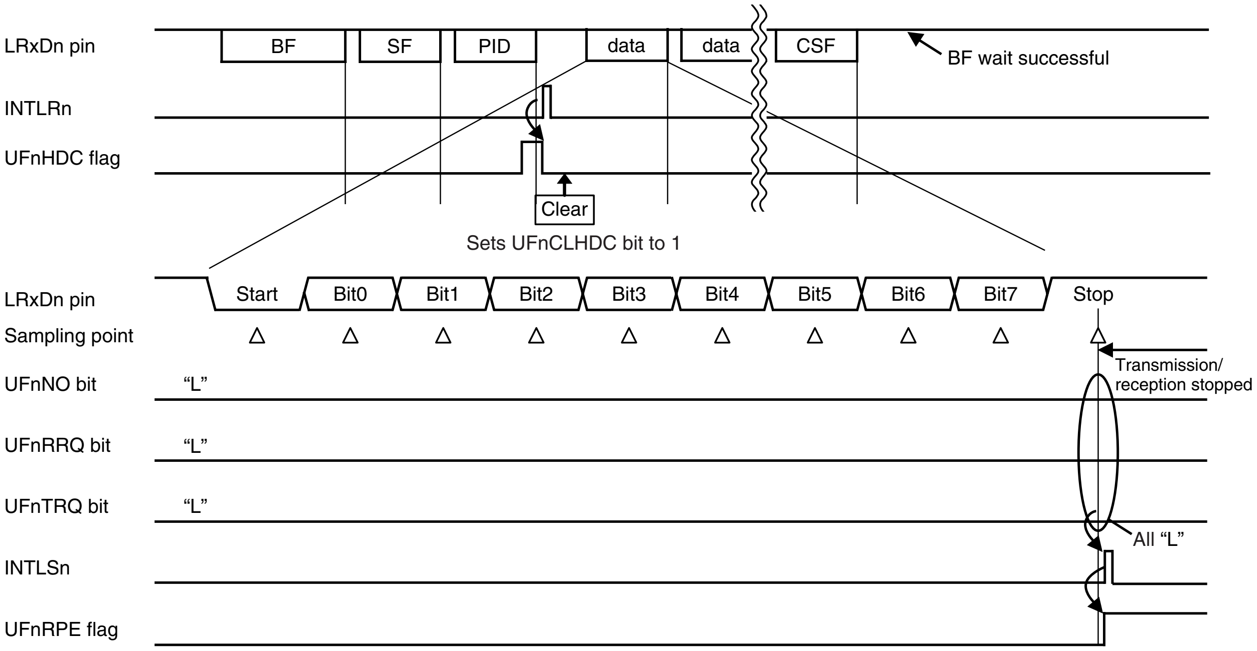
Remark n = 0, 1

12.7.2 Response preparation error detection function

If response preparation (setting of the UFnNO, UFnRRQ, and UFnTRQ bits) is not performed before reception of the first byte by a response is completed (sampling point of the stop bit (first bit)) when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B), a response preparation error flag (UFnRPE) is set, a status interrupt request signal (INTLSn) is generated, and subsequent transmission and reception processing are stopped (responses are ignored) without data being stored.

<R> When response transmission is started (UFnTRQ = 1) after reception at the LRxDn pin has been started, recognition can be performed by the occurrence of consistency errors.

Figure 12-62. Response Preparation Error Occurrence Example



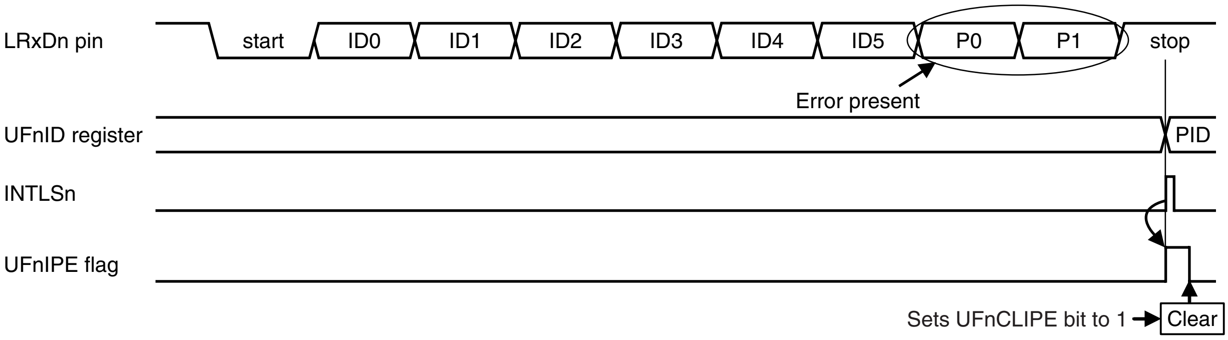
Caution If UFnCON = 0, no response preparation error will occur, because a BF reception wait state is entered after communication of the number of bytes set using the UFnBUL3 to UFnBUL0 bits is completed.
If UFnCON = 1, a response preparation error check state is entered again after communication of the number of bytes set using the UFnBUL3 to UFnBUL0 bits is completed.
A response preparation error will occur if a receive operation is started before setting UFnTRQ the next time after response transmission is completed.

Remark n = 0, 1

12.7.3 ID parity check function

When the ID parity check select bit is set (UFnIPCS = 1) in automatic baud rate mode (UFnMD1, UFnMD0 = 11B), the PID parity bits (P0, P1) are checked when the received PID is stored into the UFnID register. At that time, if either parity bit includes an error, an ID parity error flag (UFnIPE) is set, a status interrupt request signal (INTLSn) is generated instead of a reception complete interrupt request signal (INTLRn), and the PID is stored into the UFnID register.

Figure 12-63. PID Parity Error Occurrence Example



Remark n = 0, 1

12.7.4 Automatic checksum function

When the automatic checksum enable bit is set (UFnACE = 1) in automatic baud rate mode (UFnMD1, UFnMD0 = 11B), a checksum is automatically calculated. Enhanced checksum (calculation targets: PID and data) and classic checksum (calculation target: only data) can be selected for each frame by using the enhanced checksum selection bit (UFnECS).

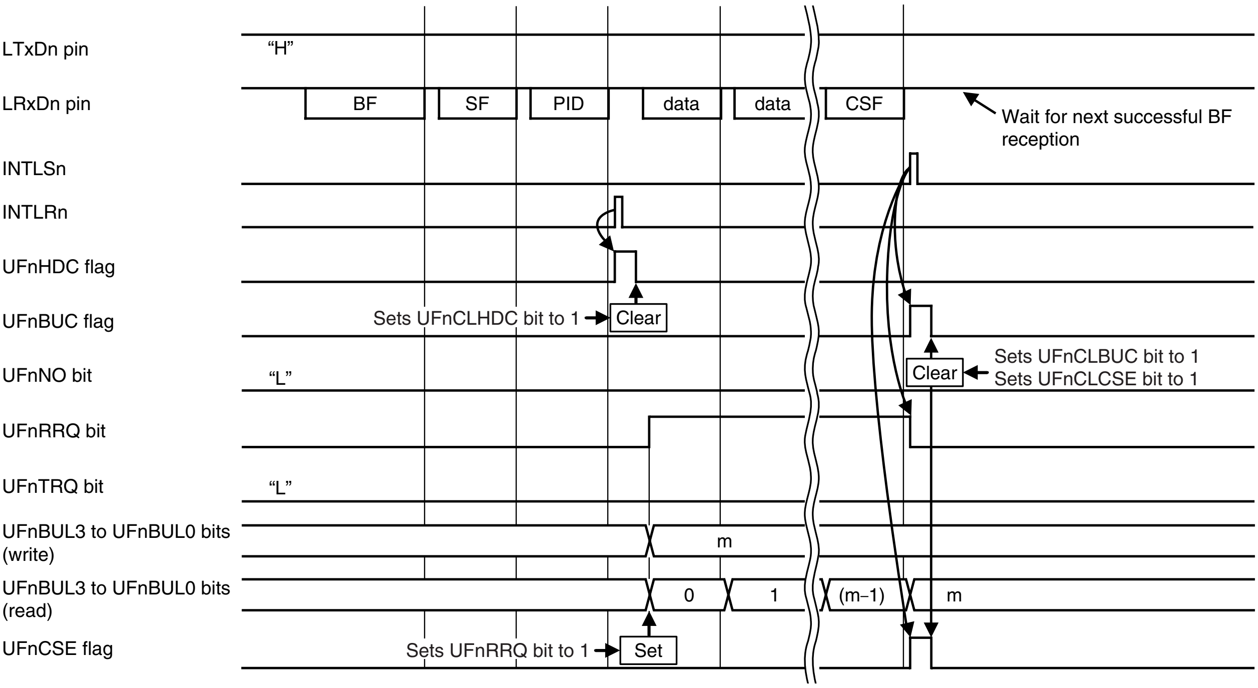
During response transmission, calculation is performed when data is transferred in 1-byte units from a buffer register to a transmit shift register[Note], and the calculation result is automatically added to the end of response transmission and transmitted. A checksum is not required to be set to a buffer by using software.

During response reception, calculation is performed when data is stored into a buffer register in 1-byte units[Note], and the stored data and calculation result are automatically compared when the received checksum is stored into a buffer. A reception complete interrupt request signal (INTLRn) is generated when the comparison result is correct. If the comparison result is illegal, however, a status interrupt request signal (INTLSn) is generated instead of a reception complete interrupt request signal (INTLRn), a checksum error flag (UFnCSE) is set, and the checksum is stored into the UFnRX register.

Note With enhanced checksum, the value of the UFnID register can be set initial value of calculate when transfer start.

Remark n = 0, 1

Figure 12-64. Automatic Checksum Error Occurrence Example (Response Reception)



Remark n = 0, 1, m = 1 to 9

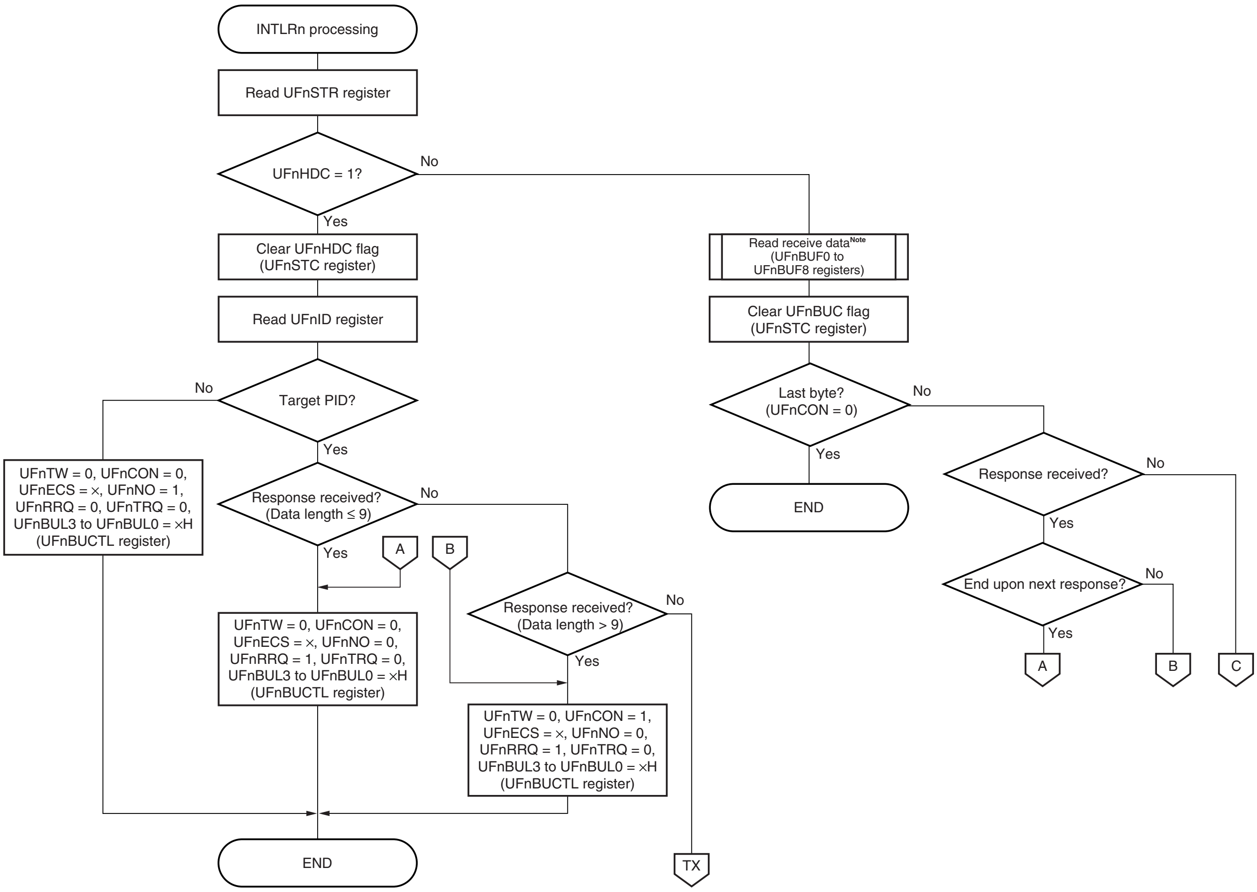
12.7.5 Multi-byte response transmission/reception function

In normal LIN communication, a response is no more than 9 bytes (including the checksum field); but in automatic baud rate mode (UFnMD1, UFnMD0 = 11B), responses of at least 10 bytes can be transmitted and received.

The processing flow of initial settings and INTLSn generation is same as the basic processing flow. See **12.7 LIN Communication Automatic Baud Rate Mode**.

The response preparation error detection function, ID parity check function, and automatic checksum function are valid.

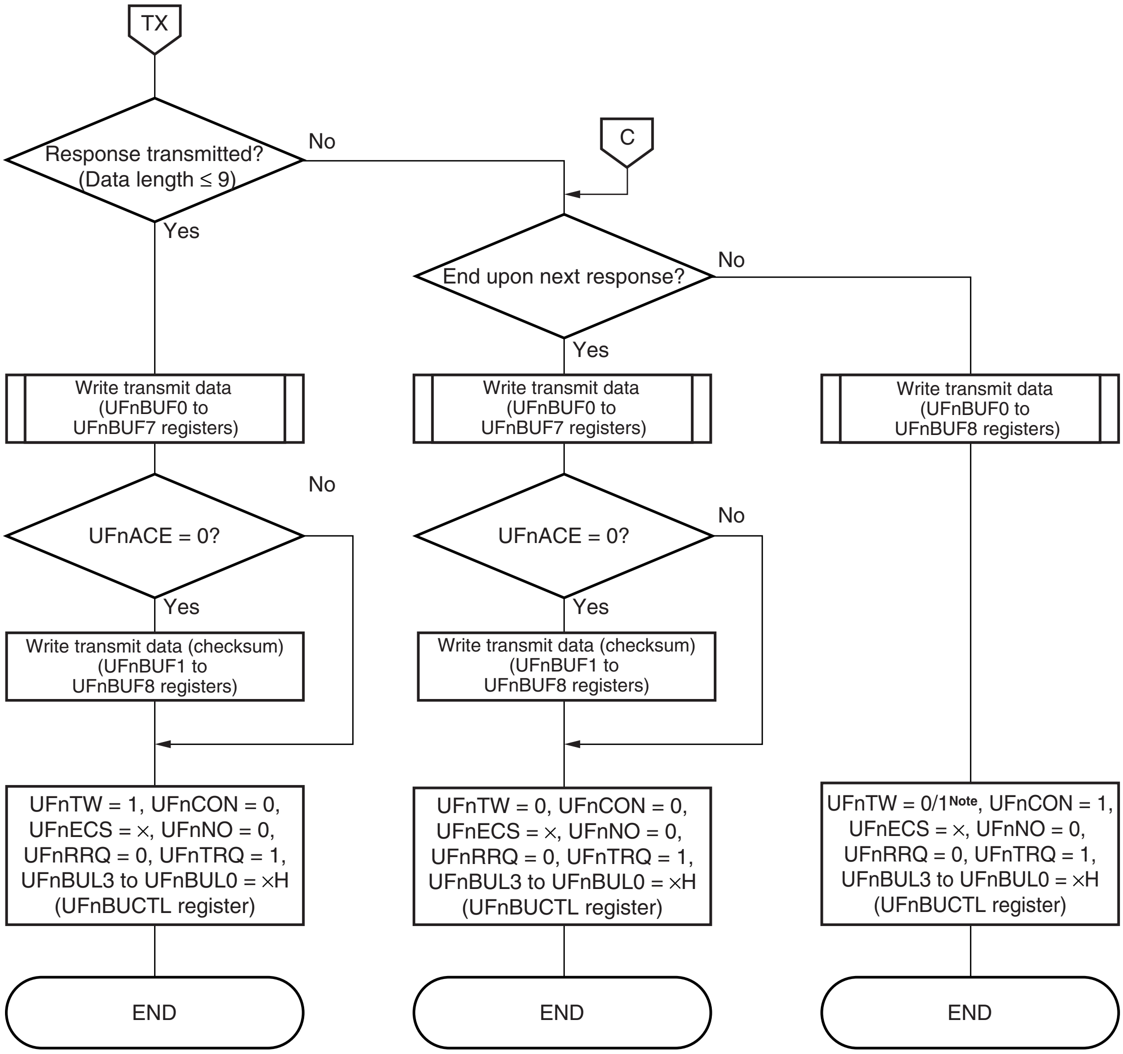
Figure 12-65. Multi-Byte Transmission/Reception Processing Flow Example (1/2)


INTLRn processing
Read UFnSTR register
UFnHDC = 1?
No
Yes
Clear UFnHDC flag (UFnSTC register)
Read UFnID register
Target PID?
No
Yes
UFnTW = 0, UFnCON = 0, UFnECS = ×, UFnNO = 1, UFnRRQ = 0, UFnTRQ = 0, UFnBUL3 to UFnBUL0 = ×H (UFnBUCTL register)
Response received? (Data length ≤ 9)
No
Yes
A
B
UFnTW = 0, UFnCON = 0, UFnECS = ×, UFnNO = 0, UFnRRQ = 1, UFnTRQ = 0, UFnBUL3 to UFnBUL0 = ×H (UFnBUCTL register)
Response received? (Data length > 9)
No
Yes
UFnTW = 0, UFnCON = 1, UFnECS = ×, UFnNO = 0, UFnRRQ = 1, UFnTRQ = 0, UFnBUL3 to UFnBUL0 = ×H (UFnBUCTL register)
END
TX
Read receive dataNote (UFnBUF0 to UFnBUF8 registers)
Clear UFnBUC flag (UFnSTC register)
Last byte? (UFnCON = 0)
No
Yes
END
Response received?
No
Yes
End upon next response?
No
Yes
A
B
C

Note This can be omitted.

Remark ×: don't care, n = 0, 1

Figure 12-65. Multi-Byte Transmission/Reception Processing Flow Example (2/2)


TX
Response transmitted?
(Data length ≤ 9)
No
Yes
C
End upon next response?
No
Yes
Write transmit data (UFnBUF0 to UFnBUF7 registers)
Write transmit data (UFnBUF0 to UFnBUF7 registers)
Write transmit data (UFnBUF0 to UFnBUF8 registers)
UFnACE = 0?
No
Yes
UFnACE = 0?
No
Yes
Write transmit data (checksum) (UFnBUF1 to UFnBUF8 registers)
Write transmit data (checksum) (UFnBUF1 to UFnBUF8 registers)
UFnTW = 1, UFnCON = 0, UFnECS = ×, UFnNO = 0, UFnRRQ = 0, UFnTRQ = 1, UFnBUL3 to UFnBUL0 = ×H (UFnBUCTL register)
UFnTW = 0, UFnCON = 0, UFnECS = ×, UFnNO = 0, UFnRRQ = 0, UFnTRQ = 1, UFnBUL3 to UFnBUL0 = ×H (UFnBUCTL register)
UFnTW = 0/1Note, UFnCON = 1, UFnECS = ×, UFnNO = 0, UFnRRQ = 0, UFnTRQ = 1, UFnBUL3 to UFnBUL0 = ×H (UFnBUCTL register)
END
END
END

Note UFnTW is set to "1" during only the first data transmission after PID reception.

Cautions 1. When the buffer length bits (UFnBUL3 to UFnBUL0) have been set to "0" or "10 to 15", reception or transmission of nine bytes is performed. When the buffer length is set to "1 to 8", buffers of the number of bytes set are used in ascending order of the buffer numbers.

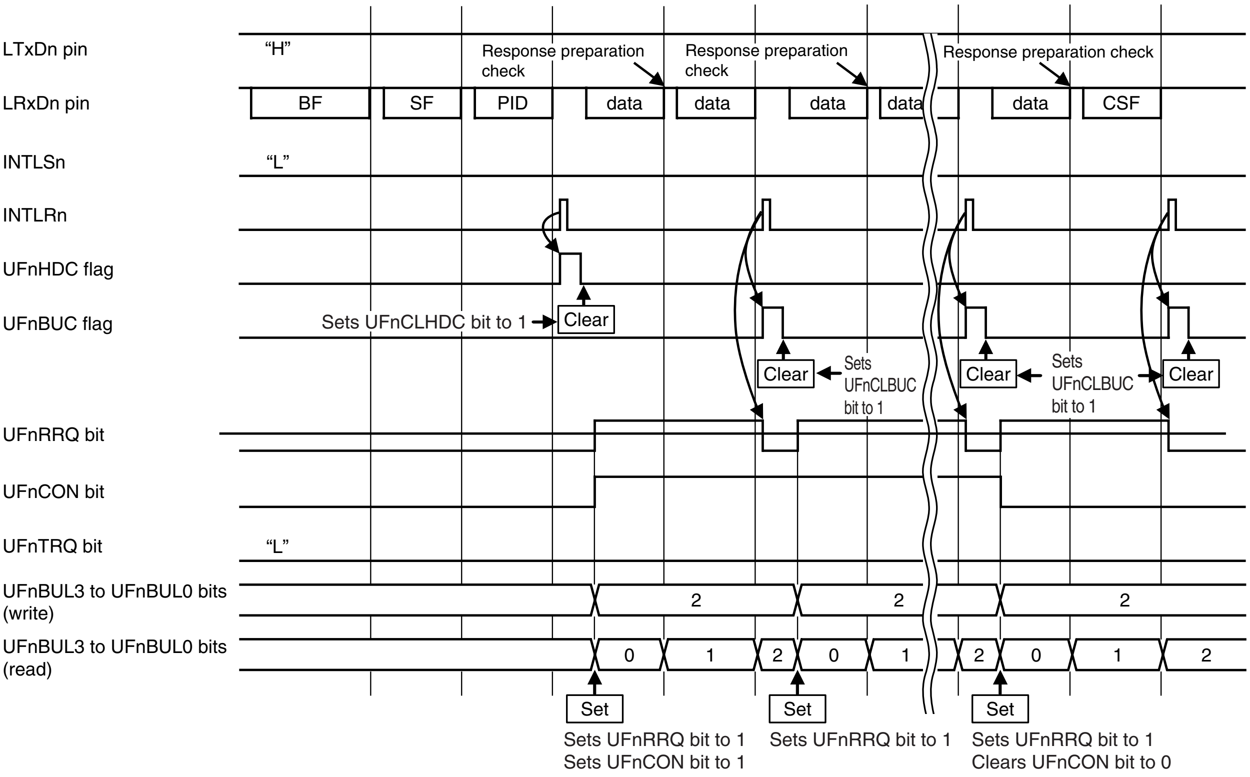
Example: When UFnBUL3 to UFnBUL0 are set to "1", data is always stored only into the UFnBUF0 register.

<R> **2. Do not set the UFnRRQ bit until reading of received data is completed, because, when the UFnRRQ bit is set, storing (overwriting) into a buffer is performed even if reading receive data has not ended.**

3. Setting the UFnTW bit is prohibited, except when operation is switched to response transmission after header reception.

Remark ×: don't care, n = 0, 1

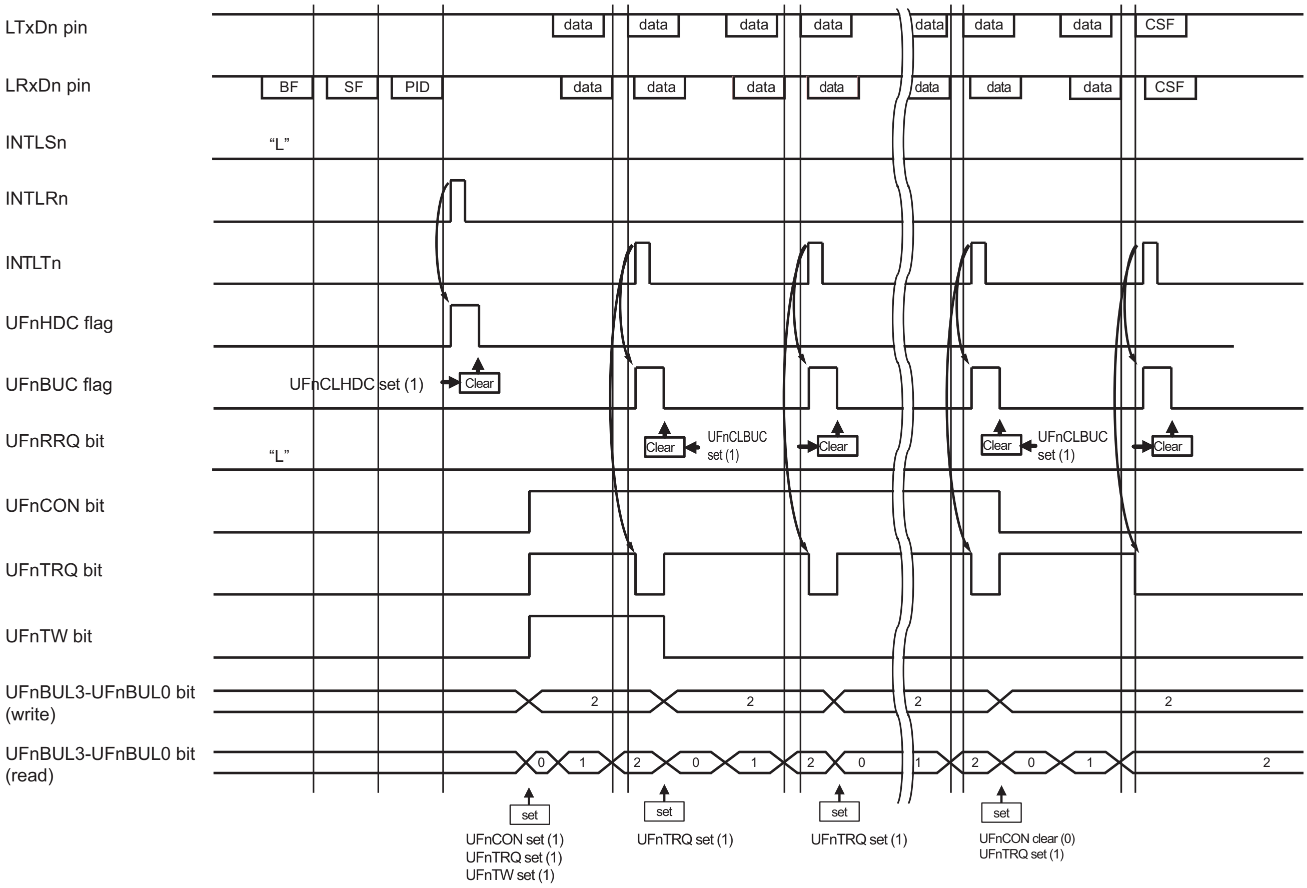
Figure 12-66. Multi-Byte Reception Implementation Example



Caution When UFnBUL3 to UFnBUL0 are "2", data is always stored into UFnBUF0 and UFnBUF1. If read processing of the receive data is not performed in time, make adjustments such as setting UFnBUL3 to UFnBUL0 to "1".

<R>

Figure 12-67. Multi-Byte Transmission Implementation Example



Caution When UFnBUL3 to UFnBUL0 are "2", data of the UFnBUF0 and UFnBUF1 bits are always transmitted and stored.

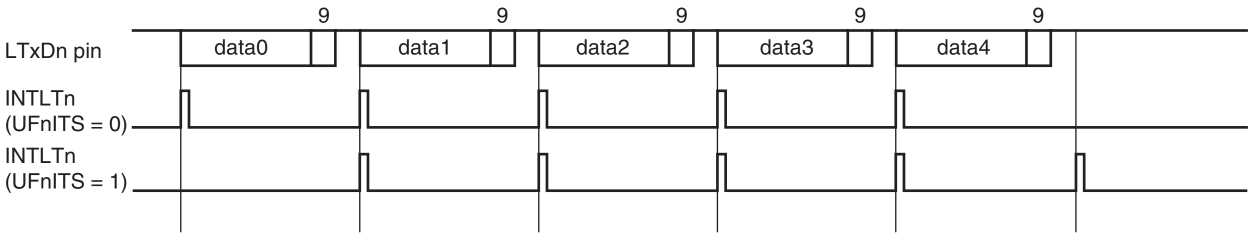
12.8 Expansion Bit Mode

When in normal UART mode (UFnMD1, UFnMD0 = 00B), data of 9-bit lengths can be transmitted or received by setting the expansion bit enable bit (UFnEBE). See **12.5.1 Data format** for the communication data format.

12.8.1 Expansion bit mode transmission

When in expansion bit mode (UFnCL = UFnEBE = 1), transmission in 9-bit lengths is started by writing 9-bit data to the UFnTX register.

Figure 12-68. Expansion Bit Mode Transmission Example (LSB First)

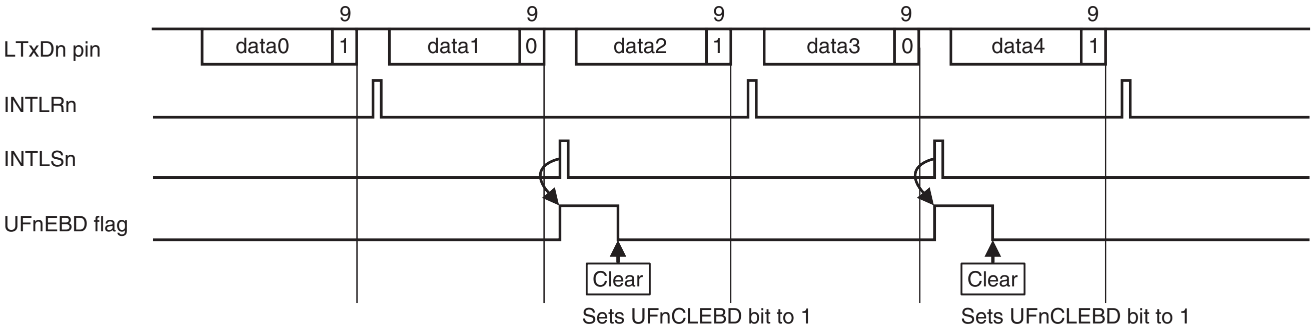


Remark n = 0, 1

12.8.2 Expansion bit mode reception (no data comparison)

When in expansion bit mode (UFnCL = UFnEBE = 1) and expansion bit data comparison is disabled (UFnEBC = 0), reception in 9-bit lengths can always be performed without data comparison. When a level set by using the expansion bit detection level select bit (UFnEBL) is detected, a status interrupt request signal (INTLSn) is generated upon completion of data reception, and an expansion bit detection flag (UFnEBD) is set. When an inverted value of the expansion bit detection level is detected, a reception complete interrupt request signal (INTLRn) is generated. In either case, the receive data is stored into the UFnRX register if no overrun error has occurred.

Figure 12-69. Expansion Bit Mode Reception (No Data Comparison) Example (LSB First, UFnEBL = 0)



Cautions 1. When a reception error (parity error, framing error, or overrun error) occurs at receive data 0, 2, or 4, a status interrupt request signal (INTLSn) is generated instead of a reception complete interrupt request signal (INTLRn), and the error flag is updated.

2. When a reception error (parity error, framing error, or overrun error) occurs at receive data 1 or 3, the error flag is also updated.

Remark n = 0, 1

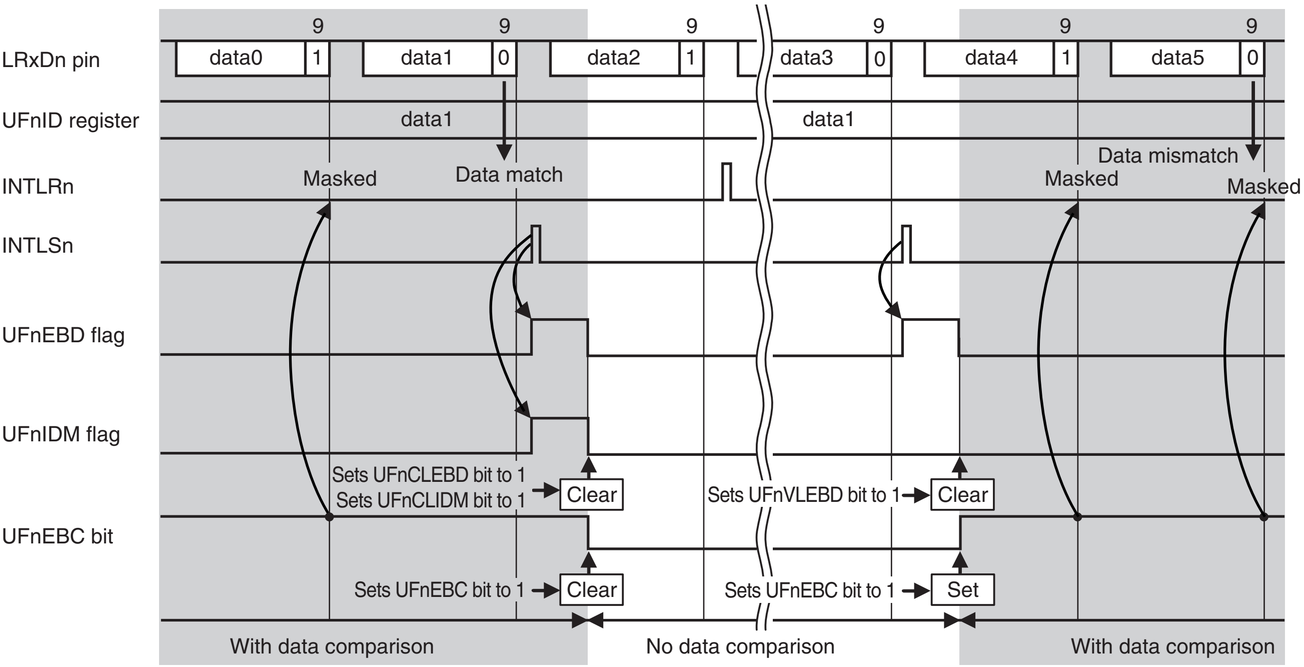
12.8.3 Expansion bit mode reception (with data comparison)

When in expansion bit mode (UFnCL = UFnEBE = 1) and expansion bit data comparison is enabled (UFnEBC = 1), if a level set by using the expansion bit detection level select bit (UFnEBL) is detected, 8 bits excluding the receive data expansion bit are compared with the value of the UFnID register set in advance.

If the comparison results have matched, a status interrupt request signal (INTLSn) is generated, an expansion bit ID match flag (UFnIDM) and an expansion bit detection flag (UFnEBD) are set, and the receive data is stored into UFnRX. If the comparison results do not match, no interrupt is generated, no flag is updated, and the receive data is not stored.

Interrupts (INTLRn, INTLSn) are generated upon all subsequent completions of data receptions and data can be received by disabling expansion bit data comparison (UFnEBC = 0) via the status interrupt servicing when the comparison results have matched. End the processing before completion of the next data reception, because data will be omitted if the UFnEBC bit is changed after the next data reception has been completed.

Figure 12-70. Expansion Bit Mode Reception (with Data Comparison) Example (LSB First, UFnEBL = 0)



Cautions 1. When a reception error (parity error, framing error, or overrun error) occurs at receive data 2, a status interrupt request signal (INTLSn) is generated instead of a reception complete interrupt request signal (INTLRn), and the error flag is updated.

2. When a reception error (parity error, framing error, or overrun error) occurs at receive data 1 or 3, the error flag is also updated. When a reception error occurs at receive data 0, 4, or 5, the error flag is not updated.

Remark n = 0, 1

12.9 Receive Data Noise Filter

The probability of malfunctioning due to noise becomes high with UART reception, because no communication clock exists. The noise filter is used to eliminate noise in a communication bus and reduce false reception of data. The noise filter becomes valid by clearing the receive data noise filter use selection bit (UFnRXFL) to "0".

A start bit and receive data input from a serial data input pin (LRxDn) are sampled with a clock (prescaler clock) divided by using a prescaler.

When the same sampling value is read twice, the match detector output changes and the receive data is sampled as the input data. Therefore, data not exceeding 2 clock width is judged to be noise and is not delivered to the internal circuit (see **Figure 12-71**). See **12.10 (1) (a) Prescaler clock (fUCLK)** regarding the base clock.

Figure 12-71. Noise Filter Circuit Example

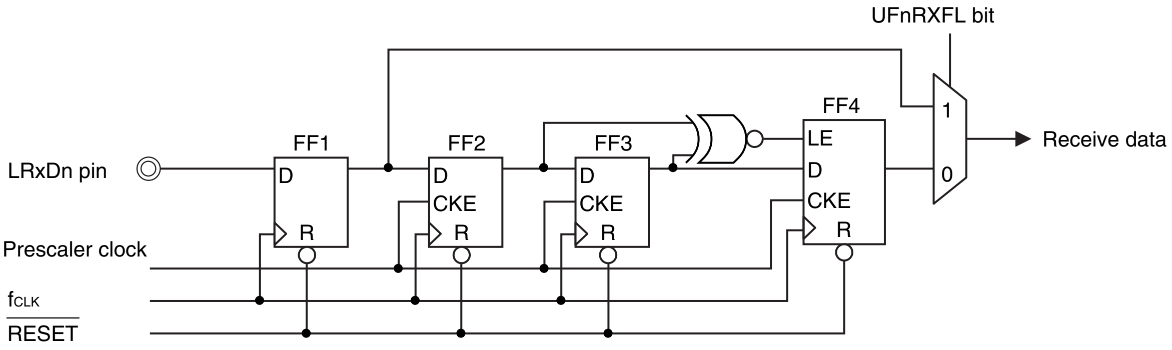
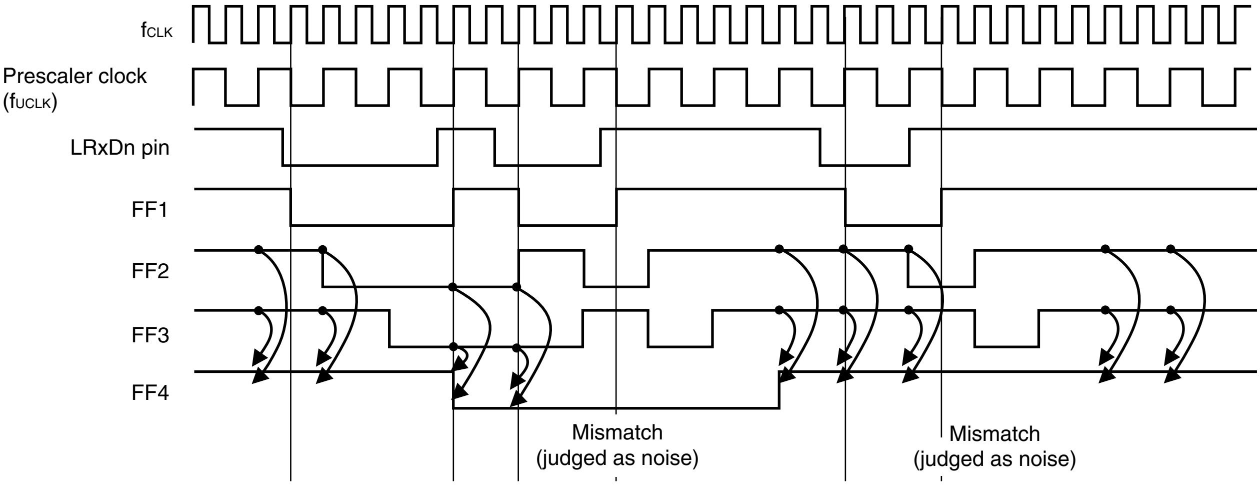


Figure 12-72. Noise Filter Timing Chart Example (UFnPRS = 1)



Remark n = 0, 1

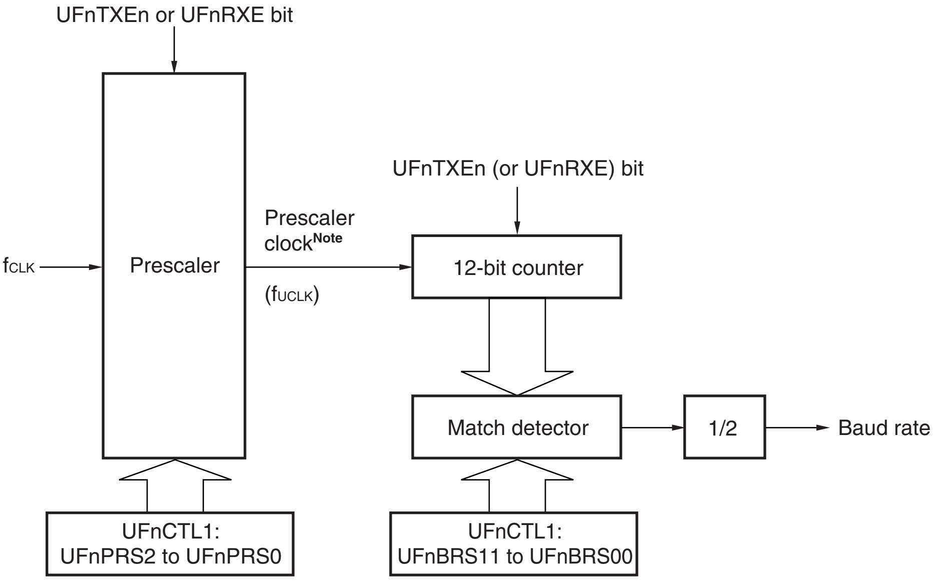
12.10 Dedicated Baud Rate Generator

The dedicated baud rate generator consists of a 3-bit prescaler block and a 12-bit programmable counter, and generates a serial clock during transmission and reception with LIN-UARTn. Regarding the serial clock, a dedicated baud rate generator output can be selected for each channel.

There is a 12-bit counter for transmission and another one for reception.

(1) Configuration of baud rate generator

Figure 12-73. Configuration of Baud Rate Generator



Note Clock that divides f_{CLK} by 1, 2, 4, 8, 16, 32, 64, or 128

<R> In automatic baud rate mode, set up a UFnRXE bit after checking that a LRxDn pin is high-level.

(a) Prescaler clock (f_{UCLK})

When the LINnEN bit of the PER register is “1”, a clock divided by a frequency division value specified by using the UFnPRS2 to UFnPRS0 bits of the UFnCTL1 register is supplied to the 12-bit counter.

This clock is called the prescaler clock and its frequency is called f_{UCLK}.

(b) Serial clock generation

A serial clock can be generated by setting the UFnCTL1 register.

The frequency division value for the 12-bit counter can be set by using the UFnBRS11 to UFnBRS00 bits of the UFnCTL1 register.

Remark n = 0, 1

(2) LIN-UARTn control register 1 (UFnCTL1)

The UFnCTL1 register is a 16-bit register that is used to control the baud rate of LIN-UARTn.

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

Reset sets this register to 0FFFH.

Figure 12-74. Format of LIN-UARTn Control Register 1 (UFnCTL1)

Address: F0242H, F0259H (UF0CTL1), F0262H, F023H (UF1UCTL1) After reset: 0000H R/W

	15	14	13	12	11	10	9	8
UFnCTL1	UFnPRS2	UFnPRS1	UFnPRS0	0	UFnBRS11	UFnBRS10	UFnBRS9	UFnBRS8
(n = 0, 1)	7	6	5	4	3	2	1	0
	UFnBRS7	UFnBRS6	UFnBRS5	UFnBRS4	UFnBRS3	UFnBRS2	UFnBRS1	UFnBRS0

UFnPRS2	UFnPRS1	UFnPRS0	Prescaler clock frequency division value
0	0	0	No division (prescaler clock = f_{CLK})
0	0	1	Division by 2 (prescaler clock = f_{CLK}/2)
0	1	0	Division by 4 (prescaler clock = f_{CLK}/4)
0	1	1	Division by 8 (prescaler clock = f_{CLK}/8)
1	0	0	Division by 16 (prescaler clock = f_{CLK}/16)
1	0	1	Division by 32 (prescaler clock = f_{CLK}/32)
1	1	0	Division by 64 (prescaler clock = f_{CLK}/64)
1	1	1	Division by 128 (prescaler clock = f_{CLK}/128)

UFn BRS11	UFn BRS10	UFn BRS9	UFn BRS8	UFn BRS7	UFn BRS6	UFn BRS5	UFn BRS4	UFn BRS3	UFn BRS2	UFn BRS1	UFn BRS0	k[Note]	Serial clock
0	0	0	0	0	0	0	0	0	0	×	×	4	f_{UCLK}/4
0	0	0	0	0	0	0	0	0	1	0	0	4	f_{UCLK}/4
0	0	0	0	0	0	0	0	0	1	0	1	5	f_{UCLK}/5
:	:	:		:		:			:	:	:	:	:
1	1	1	1	1	1	1	1	1	1	1	0	4094	f_{UCLK}/4094
1	1	1	1	1	1	1	1	1	1	1	1	4095	f_{UCLK}/4095

Note Specified value

Cautions 1. Rewriting can be performed only when the UFnTXE and UFnRXE bits of the UFnCTL0 register are "0".

2. The baud rate is the value that results by further dividing the serial clock by 2.

3. Writing to UFnBRS11 to UFnBRS00 is invalid when in automatic baud rate mode.

Remarks 1. f_{UCLK} is the frequency division value of the prescaler clock selected by using the UFnPRS2 to UFnPRS0 bits.

2. In automatic baud rate mode (UFnMD1, UFnMD0 = 11B), the value after the baud rate has been set can be checked by reading UFnBRS11 to UFnBRS00 after header reception.

3. ×: don't care

(3) Baud rate

The baud rate is obtained by the following equation.

$$\text{Baud rate} = \frac{f_{UCLK}}{2 \times k} \text{ [bps]}$$

f_{UCLK} = Frequency of prescaler clock selected by the UFnPRS2 to UFnPRS0 bits of the UFnCTL1 register
k = Value set by using the UFnBRS11 to UFnBRS00 bits of the UFnCTL1 register (k = 4, 5, 6, …, 4095)

(4) Baud rate error

The baud rate error is obtained by the following equation.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1\right) \times 100 \text{ [\%]}$$

Cautions 1. The baud rate error during transmission must be within the error tolerance on the receiving side.

2. The baud rate error during reception must satisfy the range indicated in (6) Allowable baud rate range during reception.

Example:
- CPU/peripheral hardware clock frequency = 24 MHz = 24,000,000 Hz
- Setting values
  f_{CLK} = 24 MHz
  Setting values of the UFnPRS2 to UFnPRS0 bits of the UFnCTL1 register = 001B ($f_{UCLK} = f_{CLK}/2$ = 12 MHz)
  Setting values of the UFnBRS11 to UFnBRS00 bits of the UFnCTL1 register = 000000100111B (k = 39)
- Target baud rate = 153,600 bps

- Baud rate = 12,000,000/(2 × 39)
  = 153,846 [bps]

- Error = (153,846/153,600 − 1) × 100
  = 0.160 [%]

Remark n = 0, 1

(5) Baud rate setting example

Table 12-5. Baud Rate Generator Setting Data
(Normal Operation, f_{CLK} = 24 MHz, UFnPRS2 to UFnPRS0 = 0 to 3)

Target Baud Rate (bps)	UFnPRS2 to UFnPRS0							
	0		1		2		3	
	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)
300	–	–	–	–	–	–	–	–
600	–	–	–	–	–	–	2500	0.00
1200	–	–	–	–	2500	0.00	1250	0.00
2400	–	–	2500	0.00	1250	0.00	625	0.00
4800	2500	0.00	1250	0.00	625	0.00	313	–0.16
9600	1250	0.00	625	0.00	313	–0.16	156	0.16
19200	625	0.00	313	–0.16	156	0.16	78	0.16
31250	384	0.00	192	0.00	96	0.00	48	0.00
38400	313	–0.16	156	0.16	78	0.16	39	0.16
76800	156	0.16	78	0.16	39	0.16	20	–2.34
128000	94	–0.27	47	–0.27	23	1.90	12	–2.34
153600	78	0.16	39	0.16	20	–2.34	10	–2.34
312500	38	1.05	19	1.05	10	–4.00	5	–4.00
1000000	12	0.00	6	0.00	–	–	–	–

<R> **Table 12-6. Baud Rate Generator Setting Data**
(Normal Operation, f_{CLK} = 24 MHz, UFnPRS2 to UFnPRS0 = 4 to 7)

Target Baud Rate (bps)	UFnPRS2 to UFnPRS0							
	4		5		6		7	
	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)
300	2500	0.00	1250	0.00	625	0.00	313	–0.16
600	1250	0.00	625	0.00	313	–0.16	156	0.16
1200	625	0.00	313	–0.16	156	0.16	78	0.16
2400	313	–0.16	156	0.16	78	0.16	39	0.16
4800	156	0.16	78	0.16	39	0.16	20	–2.34
9600	78	0.16	39	0.16	20	–2.34	10	–2.34
19200	39	0.16	20	–2.34	10	–2.34	5	–2.34
31250	24	0.00	12	0.00	6	0.00	–	–
38400	20	–2.34	10	–2.34	5	–2.34	–	–
76800	10	–2.34	5	–2.34	–	–	–	–
128000	6	–2.34	–	–	–	–	–	–
153600	5	–2.34	–	–	–	–	–	–
312500	–	–	–	–	–	–	–	–
1000000	–	–	–	–	–	–	–	–

Table 12-7. Baud Rate Generator Setting Data
(Normal Operation, f_{CLK} = 12 MHz, UFnPRS2 to UFnPRS0 = 0 to 3)

Target Baud Rate (bps)	UFnPRS2 to UFnPRS0							
	0		1		2		3	
	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)
300	–	–	–	–	–	–	2500	0.00
600	–	–	–	–	2500	0.00	1250	0.00
1200	–	–	2500	0.00	1250	0.00	625	0.00
2400	2500	0.00	1250	0.00	625	0.00	313	–0.16
4800	1250	0.00	625	0.00	313	–0.16	156	0.16
9600	625	0.00	313	–0.16	156	0.16	78	0.16
19200	313	–0.16	156	0.16	78	0.16	39	0.16
31250	192	0.00	96	0.00	48	0.00	24	0.00
38400	156	0.16	78	0.16	39	0.16	20	–2.34
76800	78	0.16	39	0.16	20	–2.34	10	–2.34
128000	47	–0.27	23	1.90	12	–2.34	6	–2.34
153600	39	0.16	20	–2.34	10	–2.34	5	–2.34
312500	19	1.05	10	–4.00	5	–4.00	–	–
1000000	6	0.00	–	–	–	–	–	–

Table 12-8. Baud Rate Generator Setting Data
(Normal Operation, f_{CLK} = 12 MHz, UFnPRS2 to UFnPRS0 = 4 to 7)

Target Baud Rate (bps)	UFnPRS2 to UFnPRS0							
	4		5		6		7	
	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)
300	1250	0.00	625	0.00	313	–0.16	156	0.16
600	625	0.00	313	–0.16	156	0.16	78	0.16
1200	313	–0.16	156	0.16	78	0.16	39	0.16
2400	156	0.16	78	0.16	39	0.16	20	–2.34
4800	78	0.16	39	0.16	20	–2.34	10	–2.34
9600	39	0.16	20	–2.34	10	–2.34	5	–2.34
19200	20	–2.34	10	–2.34	5	–2.34	–	–
31250	12	0.00	6	0.00	–	–	–	–
38400	10	–2.34	5	–2.34	–	–	–	–
76800	5	–2.34	–	–	–	–	–	–
128000	–	–	–	–	–	–	–	–
153600	–	–	–	–	–	–	–	–
312500	–	–	–	–	–	–	–	–
1000000	–	–	–	–	–	–	–	–

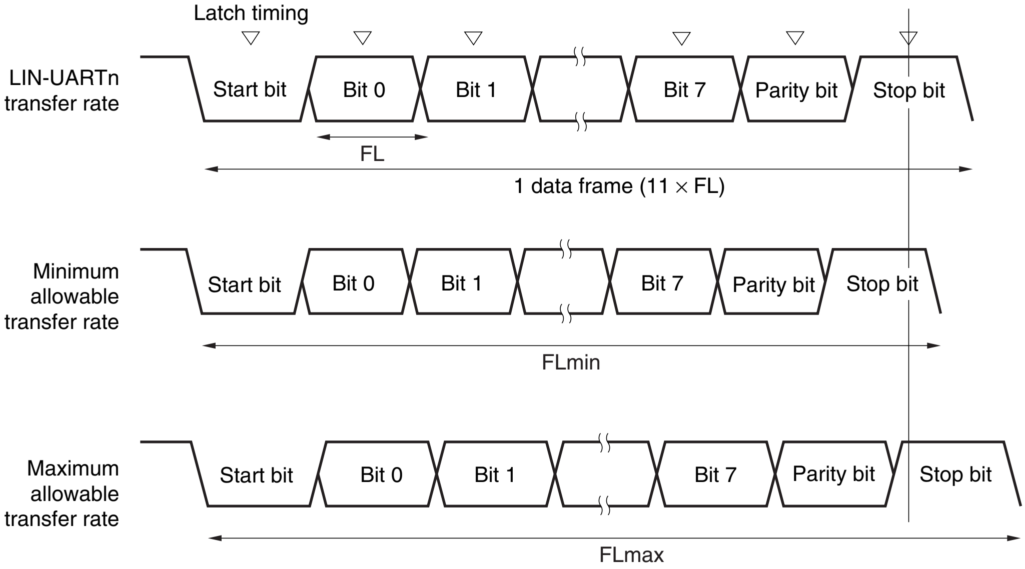
(6) Allowable baud rate range during reception

The baud rate error range at the destination that is allowable during reception is shown below.

The 11-bit reception which added the data bit length 8 bits, the start bit, the stop bit, and the parity bit is shown in an example.

Caution The baud rate error during reception must be set within the allowable error range using the following equation.

Figure 12-75. Allowable Baud Rate Range During Reception



As shown in Figure 12-75, the receive data latch timing is determined by the counter set using the UFnCTL1 register following start bit detection. The transmit data can be normally received if up to the last data (stop bit) can be received in time for this latch timing.

When this is applied to 11-bit reception while the data bit length is 8 bits, the following is the theoretical result.

$FL = (Brate)^{-1}$

Brate: LIN-UARTn baud rate
k: Setting value of UFnCTL1
FL: 1-bit data length
Latch timing margin: 2 clocks

Minimum allowable transfer rate: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum baud rate that can be received by the destination is as follows.

$$\text{BRmax} = (\text{FLmin}/11)^{-1} = \frac{22k}{21k + 2}\ \text{Brate}$$

Similarly, obtaining the following maximum allowable transfer rate yields the following.

$$\frac{10}{11} \times \text{FLmax} = 11 \times \text{FL} - \frac{k + 2}{2 \times k} \times \text{FL} = \frac{21k - 2}{2 \times k}\ \text{FL}$$

$$\text{FLmax} = \frac{21k - 2}{20\ k}\ \text{FL} \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$\text{BRmin} = (\text{FLmax}/11)^{-1} = \frac{20k}{21k - 2}\ \text{Brate}$$

Table 12-9 shows the calculation result of the allowable baud rate error between LIN-UARTn and transmitting origin by the formula of the baud rate value of the above-mentioned minimum and maximum.

<R>

Table 12-9. Maximum/Minimum Allowable Baud Rate Error

Division Ratio (k)	Maximum Allowable Baud Rate Error			Minimum Allowable Baud Rate Error		
	BN = 9	BN = 11	BN = 12	BN = 9	BN = 11	BN = 12
4	+2.85%	+2.32%	+2.12%	–3.03%	–2.43%	–2.22%
8	+4.34%	+3.52%	+3.22%	–4.47%	–3.61%	–3.29%
16	+5.10%	+4.14%	+3.78%	–5.18%	–4.19%	–3.82%
64	+5.68%	+4.60%	+4.20%	–5.70%	–4.61%	–4.21%
128	+5.78%	+4.68%	+4.27%	–5.79%	–4.69%	–4.28%
256	+5.83%	+4.72%	+4.31%	–5.83%	–4.72%	–4.31%
512	+5.85%	+4.74%	+4.33%	–5.86%	–4.74%	–4.33%
1024	+5.87%	+4.75%	+4.33%	–5.87%	–4.75%	–4.33%
2048	+5.87%	+4.75%	+4.34%	–5.87%	–4.75%	–4.34%
4095	+3.42%	+4.75%	+4.34%	–3.59%	–4.75%	–4.34%

Remarks 1. The reception accuracy depends on the bit count in 1 frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the accuracy.

2. BN: Number of bits from the start bit to the stop bit
K: Setting values of UFnCTL1.UFnBRS[11:0]

12.11 Cautions for Use

(1) Execute a STOP instruction during a LIN-UART operation after stopping LIN-UART.

(2) Start up the LIN-UARTn in the following sequence.
<1> Set the ports.
<2> Set PER0.LINnEN to 1.
<3> Set UFnCTL0.UFnTXE to 1, and UFnCTL0.UFnRXE to 1.

(3) Stop the LIN-UARTn in the following sequence.
<1> Set UFnCTL0.UFnTXE to 0, and UFnCTL0.UFnRXE to 0.
<2> Set PER1.LINnEN to 0.
<3> Set the ports. (It is not a problem if port setting is not changed.)

(4) In transmit mode (UFnCTL0.UFnTXE = 1), do not overwrite the same value to the UFnTX register by software because transmission starts by writing to this register. To transmit the same value continuously, overwrite the same value.

Remark n = 0, 1

CHAPTER 13 CAN CONTROLLER

13.1 Outline Description

This product features an on-chip 1-channel CAN (Controller Area Network) controller that complies with CAN protocol as standardized in ISO 11898.

13.1.1 Features

- Compliant with ISO 11898 and tested according to ISO/DIS 16845 (CAN conformance test)
- Standard frame and extended frame transmission/reception enabled
- Transfer rate: 1 Mbps max. (CAN input clock ≥ 8 MHz)
- 16 message buffers/1 channel
- Receive/transmit history list function
- Automatic block transmission function
- Multi-buffer receive block function
- Mask setting of four patterns is possible for each channel

13.1.2 Overview of functions

Table 13-1 presents an overview of the CAN controller functions.

Table 13-1. Overview of Functions

Function	Details
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Maximum 1 Mbps (CAN clock input ≥ 8 MHz)
Data storage	Storing messages in the CAN RAM
Number of messages	- 16 message buffers/1 channel - Each message buffer can be set to be either a transmit message buffer or a receive message buffer.
Message reception	- Unique ID can be set to each message buffer. - Mask setting of four patterns is possible for each channel. - A receive completion interrupt is generated each time a message is received and stored in a message buffer. - Two or more receive message buffers can be used as a FIFO receive buffer (multi-buffer receive block function). - Receive history list function
Message transmission	- Unique ID can be set to each message buffer. - Transmit completion interrupt for each message buffer - Message buffer number 0 to 7 specified as the transmit message buffer can be used for automatic block transfer. Message transmission interval is programmable (automatic block transmission function (hereafter referred to as “ABT”)). - Transmission history list function
Remote frame processing	Remote frame processing by transmit message buffer
Time stamp function	- The time stamp function can be set for a message reception when a 16-bit timer is used in combination. Time stamp capture trigger can be selected (SOF or EOF in a CAN message frame can be detected.).
Diagnostic function	- Readable error counters - “Valid protocol operation flag” for verification of bus connections - Receive-only mode - Single-shot mode - CAN protocol error type decoding - Self-test mode
Release from bus-off state	- Forced release from bus-off (by ignoring timing constraint) possible by software. - No automatic release from bus-off (software must re-enable).
Power save mode	- CAN sleep mode (can be woken up by CAN bus) - CAN stop mode (cannot be woken up by CAN bus)

13.1.3 Configuration

The CAN controller is composed of the following four blocks.

(1) Interface

This functional block provides an internal bus interface and means of transmitting and receiving signals between the CAN module and the host CPU.

(2) MCM (Memory Control Module)

This functional block controls access to the CAN protocol layer and to the CAN RAM within the CAN module.

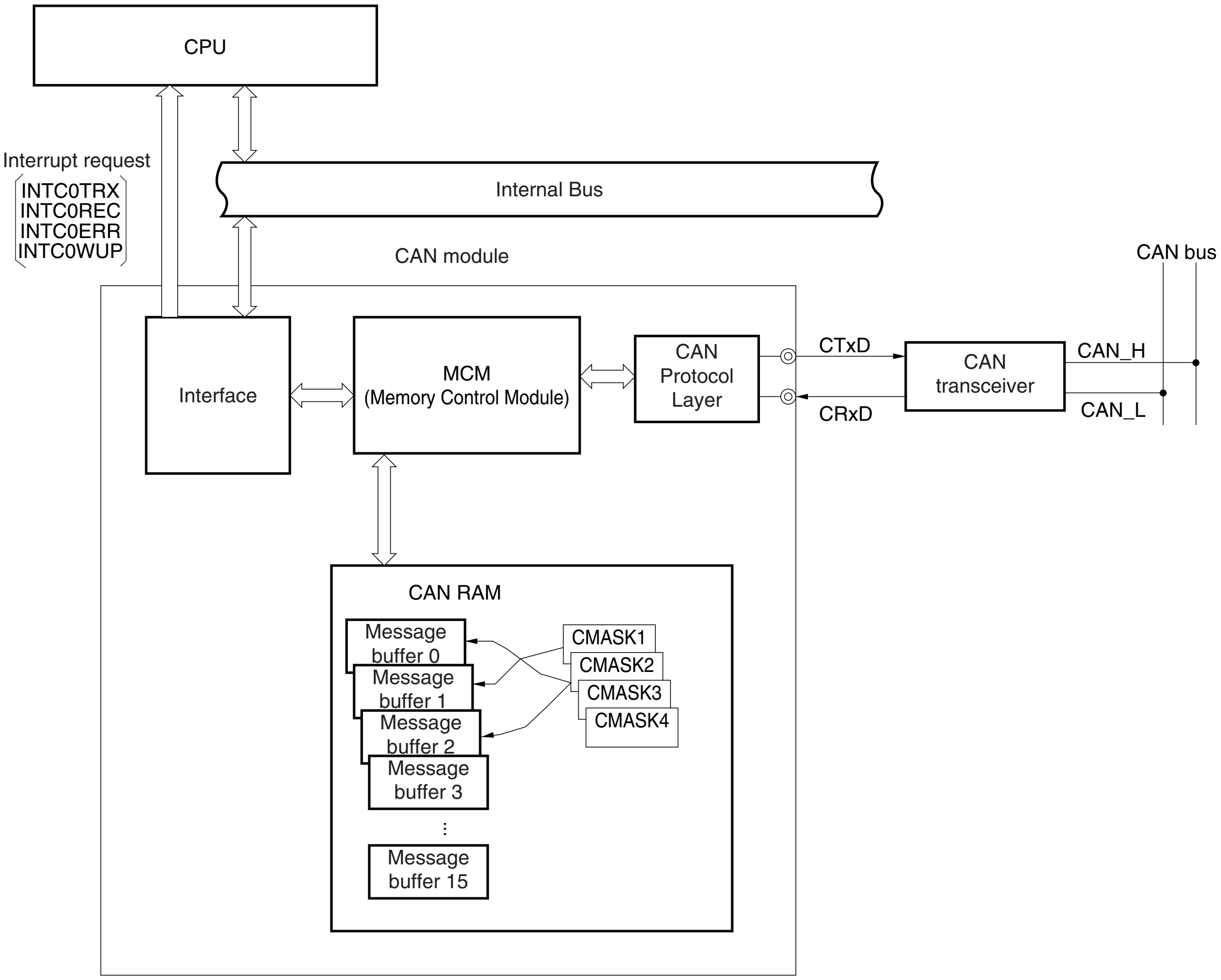
(3) CAN protocol layer

This functional block is involved in the operation of the CAN protocol and its related settings.

(4) CAN RAM

This is the CAN memory functional block, which is used to store message IDs, message data, etc.

Figure 13-1. Block Diagram of CAN Module



13.2 CAN Protocol

CAN (Controller Area Network) is a high-speed multiplex communication protocol for real-time communication in automotive applications (class C). CAN is prescribed by ISO 11898. For details, refer to the ISO 11898 specifications.

The CAN specification is generally divided into two layers: a physical layer and a data link layer. In turn, the data link layer includes logical link and medium access control. The composition of these layers is illustrated below.

Figure 13-2. Composition of Layers

Higher ↕ Lower

Data link layer[Note]	• Logical link control (LLC)	• Acceptance filtering • Overload report • Recovery management
	• Medium access control (MAC)	• Data capsuled/not capsuled • Frame coding (stuffing/not stuffing) • Medium access management • Error detection • Error report • Acknowledgement • Seriated/not seriated
Physical layer		Prescription of signal level and bit description

Note CAN controller specification

13.2.1 Frame format

(1) Standard format frame

- The standard format frame uses 11-bit identifiers, which means that it can handle up to 2048 messages.

(2) Extended format frame

- The extended format frame uses 29-bit (11 bits + 18 bits) identifiers which increase the number of messages that can be handled to 2048 × 218 messages.
- Extended format frame is set when “recessive level” (CMOS level equals “1”) is set for both the SRR and IDE bits in the arbitration field.

13.2.2 Frame types

The following four types of frames are used in the CAN protocol.

Table 13-2. Frame Types

Frame Type	Description
Data frame	Frame used to transmit data
Remote frame	Frame used to request a data frame
Error frame	Frame used to report error detection
Overload frame	Frame used to delay the next data frame or remote frame

(1) Bus value

The bus values are divided into dominant and recessive.

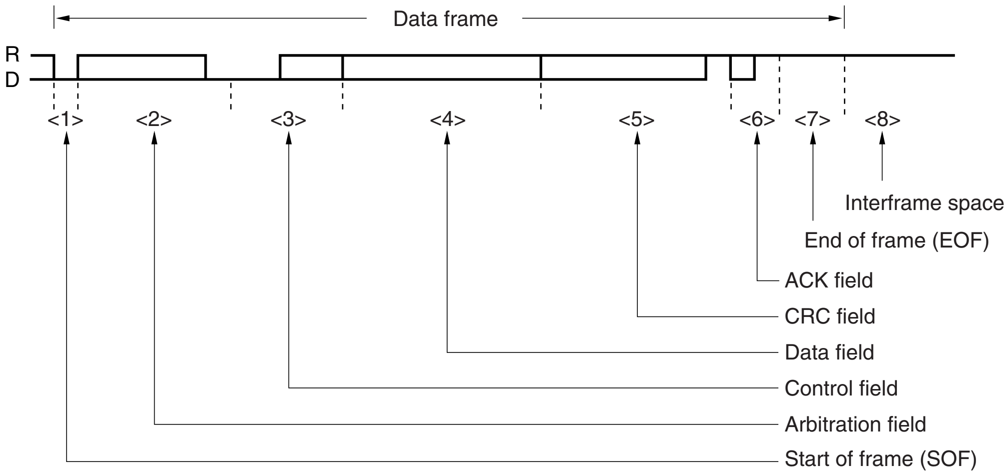
- Dominant level is indicated by logical 0.
- Recessive level is indicated by logical 1.
- When a dominant level and a recessive level are transmitted simultaneously, the bus value becomes dominant level.

13.2.3 Data frame and remote frame

(1) Data frame

A data frame is composed of seven fields.

Figure 13-3. Data Frame

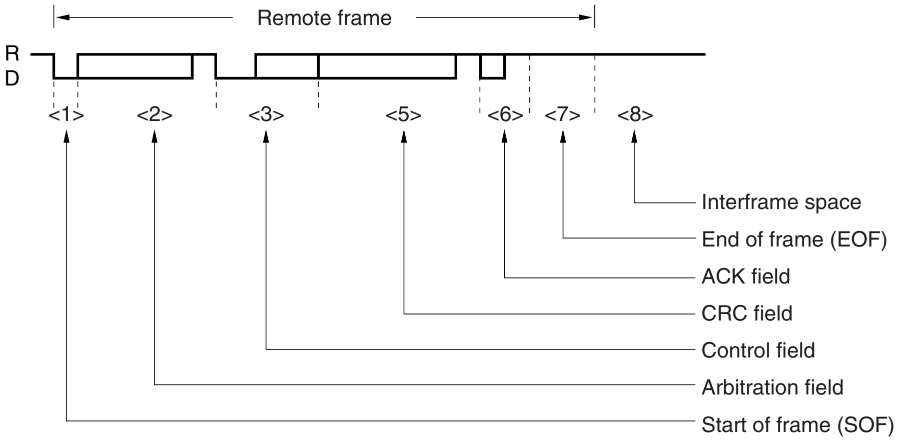


Remark D: Dominant = 0
R: Recessive = 1

(2) Remote frame

A remote frame is composed of six fields.

Figure 13-4. Remote Frame



Remarks 1. The data field is not transferred even if the control field’s data length code is not “0000B”.

2. D: Dominant = 0

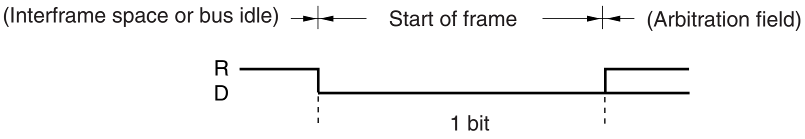
R: Recessive = 1

(3) Description of fields

<1> Start of frame (SOF)

The start of frame field is located at the start of a data frame or remote frame.

Figure 13-5. Start of Frame (SOF)



Remark D: Dominant = 0

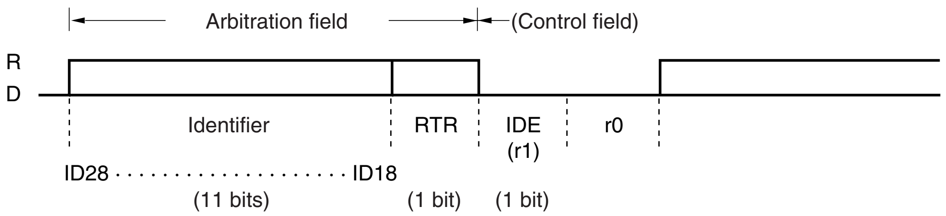
R: Recessive = 1

- If dominant level is detected in the bus idle state, a hard-synchronization is performed (the current TQ is assigned to be the SYNC segment).
- If dominant level is sampled at the sample point following such a hard-synchronization, the bit is assigned to be a SOF. If recessive level is detected, the protocol layer returns to the bus idle state and regards the preceding dominant pulse as a disturbance only. No error frame is generated in such case.

<2> **Arbitration field**

The arbitration field is used to set the priority, data frame/remote frame, and frame format.

Figure 13-6. Arbitration Field (in Standard Format Mode)



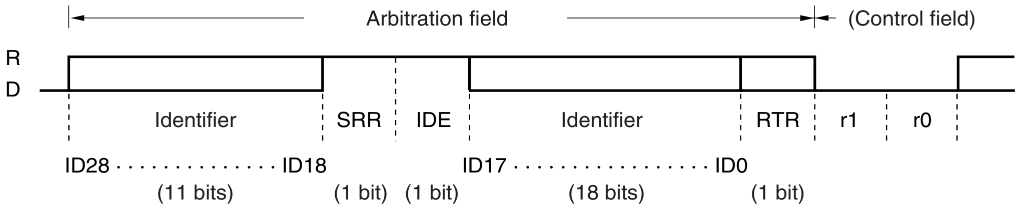
Cautions 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Remark D: Dominant = 0

R: Recessive = 1

Figure 13-7. Arbitration Field (in Extended Format Mode)



Cautions 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Remark D: Dominant = 0

R: Recessive = 1

Table 13-3. RTR Frame Settings

Frame Type	RTR Bit
Data frame	0 (D)
Remote frame	1 (R)

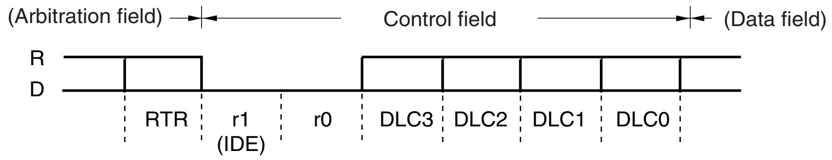
Table 13-4. Frame Format Setting (IDE Bit) and Number of Identifier (ID) Bits

Frame Format	SRR Bit	IDE Bit	Number. of Bits
Standard format mode	None	0 (D)	11 bits
Extended format mode	1 (R)	1 (R)	29 bits

<3> Control field

The control field sets “DLC” as the number of data bytes in the data field (DLC = 0 to 8).

Figure 13-8. Control Field



Remark D: Dominant = 0
R: Recessive = 1

In a standard format frame, the control field’s IDE bit is the same as the r1 bit.

Table 13-5. Data Length Setting

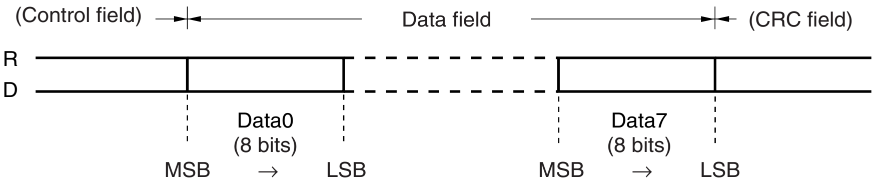
Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than above				8 bytes regardless of the value of DLC3 to DLC0

Caution In the remote frame, there is no data field even if the data length code is not 0000B.

<4> Data field

The data field contains the amount of data (byte units) set by the control field. Up to 8 units of data can be set.

Figure 13-9. Data Field

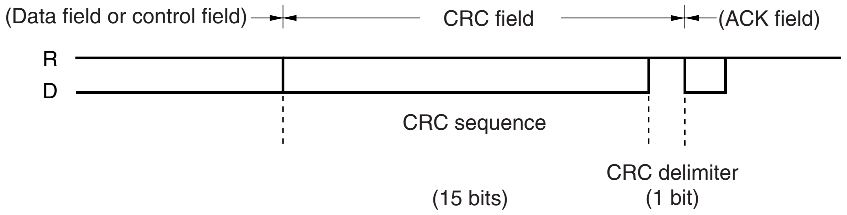


Remark D: Dominant = 0
R: Recessive = 1

<5> CRC field

The CRC field is a 16-bit field that is used to check for errors in transmit data.

Figure 13-10. CRC Field



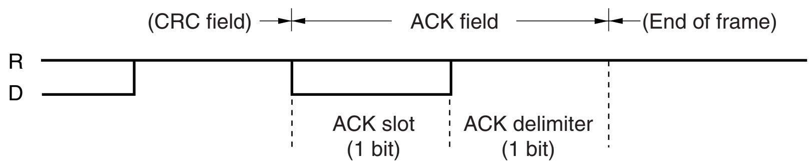
Remark D: Dominant = 0
R: Recessive = 1

- The polynomial P(X) used to generate the 15-bit CRC sequence is expressed as follows.
  $P(X) = X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$
- Transmitting node: Transmits the CRC sequence calculated from the data (before bit stuffing) in the start of frame, arbitration field, control field, and data field.
- Receiving node: Compares the CRC sequence calculated using data bits that exclude the stuffing bits in the receive data with the CRC sequence in the CRC field. If the two CRC sequences do not match, the node issues an error frame.

<6> ACK field

The ACK field is used to acknowledge normal reception.

Figure 13-11. ACK Field



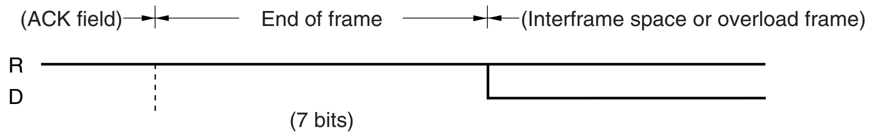
Remark D: Dominant = 0
R: Recessive = 1

- If no CRC error is detected, the receiving node sets the ACK slot to the dominant level.
- The transmitting node outputs two recessive-level bits.

<7> End of frame (EOF)

The end of frame field indicates the end of data frame/remote frame.

Figure 13-12. End of Frame (EOF)



Remark D: Dominant = 0
R: Recessive = 1

<8> **Interframe space**

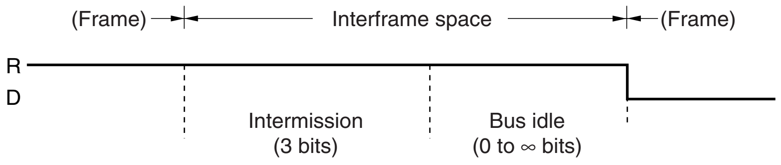
The interframe space is inserted after a data frame, remote frame, error frame, or overload frame to separate one frame from the next.

- The bus state differs depending on the error status.

(a) Error active node

The interframe space consists of a 3-bit intermission field and a bus idle field.

Figure 13-13. Interframe Space (Error Active Node)

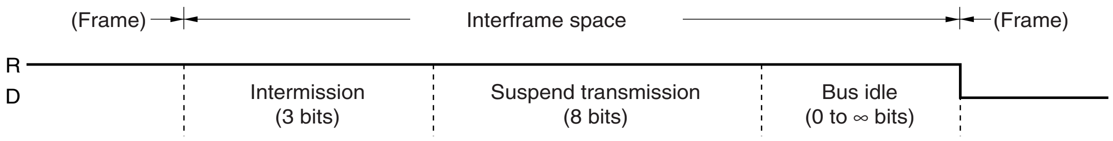


Remarks 1. Bus idle: State in which the bus is not used by any node.
2. D: Dominant = 0
R: Recessive = 1

(b) Error passive node

The interframe space consists of an intermission field, a suspend transmission field, and a bus idle field.

Figure 13-14. Interframe Space (Error Passive Node)



Remarks 1. Bus idle: State in which the bus is not used by any node.
Suspend transmission: Sequence of 8 recessive-level bits transmitted from the node in the error passive status.
2. D: Dominant = 0
R: Recessive = 1

Usually, the intermission field is 3 bits. If the transmitting node detects a dominant level at the third bit of the intermission field, however, it executes transmission.

- Operation in error status

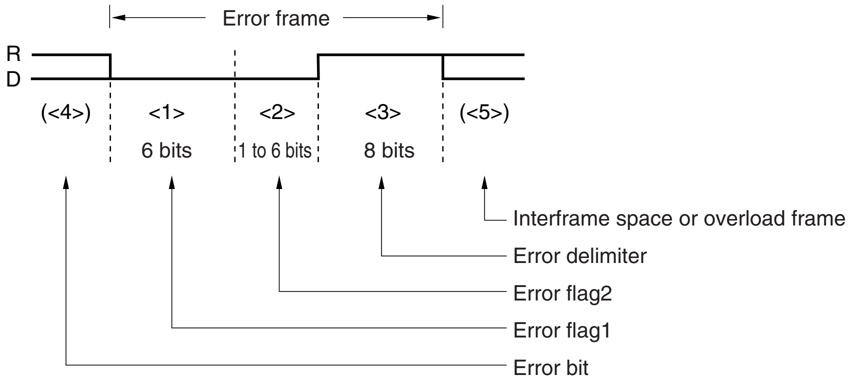
Table 13-6. Operation in Error Status

Error Status	Operation
Error active	A node in this status can transmit immediately after a 3-bit intermission.
Error passive	A node in this status can transmit 8 bits after the intermission.

13.2.4 Error frame

An error frame is output by a node that has detected an error.

Figure 13-15. Error Frame



Remark D: Dominant = 0
R: Recessive = 1

Table 13-7. Definition Error Frame Fields

No.	Name	Bit Count	Definition
<1>	Error flag1	6	Error active node: Outputs 6 dominant-level bits consecutively. Error passive node: Outputs 6 recessive-level bits consecutively. If another node outputs a dominant level while one node is outputting a passive error flag, the passive error flag is not cleared until the same level is detected 6 bits in a row.
<2>	Error flag2	0 to 6	Nodes receiving error flag 1 detect bit stuff errors and issues this error flag.
<3>	Error delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Error bit	–	The bit at which the error was detected. The error flag is output from the bit next to the error bit. In the case of a CRC error, this bit is output following the ACK delimiter.
<5>	Interframe space/overload frame	–	An interframe space or overload frame starts from here.5

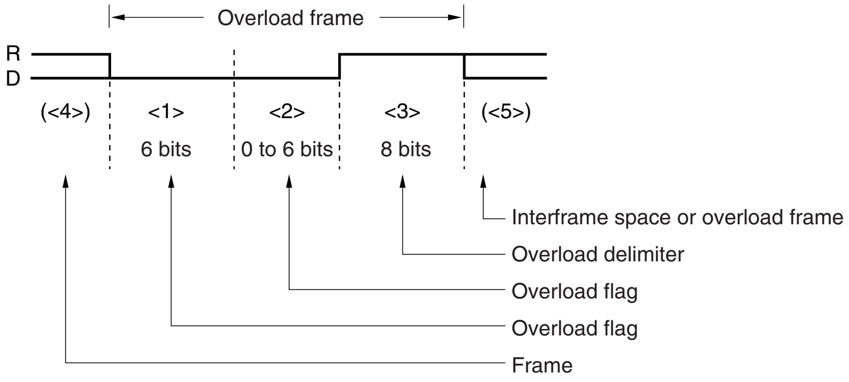
13.2.5 Overload frame

An overload frame is transmitted under the following conditions.

- When the receiving node has not completed the reception operation[Note]
- If a dominant level is detected at the first two bits during intermission
- If a dominant level is detected at the last bit (7th bit) of the end of frame or at the last bit (8th bit) of the error delimiter/overload delimiter

Note The CAN is internally fast enough to process all received frames not generating overload frames.

Figure 13-16. Overload Frame



Remark D: Dominant = 0
R: Recessive = 1

Table 13-8. Definition of Overload Frame Fields

No	Name	Bit Count	Definition
<1>	Overload flag	6	Outputs 6 dominant-level bits consecutively.
<2>	Overload flag from other node	0 to 6	The node that received an overload flag in the interframe space outputs an overload flag.
<3>	Overload delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Frame	–	Output following an end of frame, error delimiter, or overload delimiter.
<5>	Interframe space/overload frame	–	An interframe space or overload frame starts from here.

13.3 Functions

13.3.1 Determining bus priority

(1) When a node starts transmission:

- During bus idle, the node that output data first transmits the data.

(2) When more than one node starts transmission:

- The node that outputs the dominant level for the longest consecutively from the first bit of the arbitration field acquires the bus priority (if a dominant level and a recessive level are simultaneously transmitted, the dominant level is taken as the bus value).
- The transmitting node compares its output arbitration field and the data level on the bus.

Table 13-9. Determining Bus Priority

Level match	Continuous transmission
Level mismatch	Stops transmission at the bit where mismatch is detected and starts reception at the following bit

(3) Priority of data frame and remote frame

- When a data frame and a remote frame are on the bus, the data frame has priority because its RTR bit, the last bit in the arbitration field, carries a dominant level.

Caution If the extended-format data frame and the standard-format remote frame conflict on the bus (if ID28 to ID18 of both of them are the same), the standard-format remote frames takes priority.

13.3.2 Bit stuffing

Bit stuffing is used to establish synchronization by appending 1-bit inverted data if the same level continues for 5 bits, in order to prevent a burst error.

Table 13-10. Bit Stuffing

Transmission	During the transmission of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, 1 inverted-level bit of data is inserted before the following bit.
Reception	During the reception of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, reception is continued after deleting the next bit.

13.3.3 Multi masters

As the bus priority (a node acquiring transmit functions) is determined by the identifier, any node can be the bus master.

13.3.4 Multi cast

Although there is one transmitting node, two or more nodes can receive the same data at the same time because the same identifier can be set to two or more nodes.

13.3.5 CAN sleep mode/CAN stop mode function

The CAN sleep mode/CAN stop mode function puts the CAN controller in waiting mode to achieve low power consumption.

The controller is woken up from the CAN sleep mode by bus operation but it is not woken up from the CAN stop mode by bus operation (the CAN stop mode is controlled by CPU access).

13.3.6 Error control function

(1) Error types

Table 13-11. Error Types

Type	Description of Error		Detection State	
	Detection Method	Detection Condition	Transmission/ Reception	Field/Frame
Bit error	Comparison of output level and level on the bus	Mismatch of levels	Transmitting/ receiving node	Bit that outputting data on the bus at the start of frame to end of frame, error frame and overload frame.
Stuff error	Check the receive data at the stuff bit	6 consecutive bits of the same output level	Receiving node	Start of frame to CRC sequence
CRC error	Comparison of the CRC sequence generated from the receive data and the received CRC sequence	Mismatch of CRC	Receiving node	CRC field
Form error	Field/frame check of the fixed format	Detection of fixed format violation	Receiving node	CRC delimiter ACK field End of frame Error frame Overload frame
ACK error	Check of the ACK slot by the transmitting node	Detection of recessive level in ACK slot	Transmitting node	ACK slot

(2) Output timing of error frame

Table 13-12. Output Timing of Error Frame

Type	Output Timing
Bit error, stuff error, form error, ACK error	Error frame output is started at the timing of the bit following the detected error.
CRC error	Error frame output is started at the timing of the bit following the ACK delimiter.

(3) Processing in case of error

The transmission node re-transmits the data frame or remote frame after the error frame (However, it does not re-transmit the frame in the single-shot mode.).

(4) Error state

(a) Types of error states

The following three types of error states are defined by the CAN specification.

- Error active
- Error passive
- Bus-off

These types of error states are classified by the values of the TEC7 to TEC0 bits (transmission error counter bits) and the REC6 to REC0 bits (reception error counter bits) of the CAN error counter register (CERC) as shown in Table 13-13.

The present error state is indicated by the CAN module information register (CINFO).

When each error counter value becomes equal to or greater than the error warning level (96), the TECS0 or RECS0 bit of the CINFO register is set to 1. In this case, the bus state must be tested because it is considered that the bus has a serious fault. An error counter value of 128 or more indicates an error passive state and the TECS1 or RECS1 bit of the CINFO register is set to 1.

- If the value of the transmission error counter is greater than or equal to 256 (actually, the transmission error counter does not indicate a value greater than or equal to 256), the bus-off state is reached and the BOFF bit of the CINFO register is set to 1.
- If only one node is active on the bus at startup (i.e., a particular case such as when the bus is connected only to the local station), ACK is not returned even if data is transmitted. Consequently, re-transmission of the error frame and data is repeated. In the error passive state, however, the transmission error counter is not incremented and the bus-off state is not reached.

Table 13-13. Types of Error States

Type	Operation	Value of Error Counter	Indication of CINFO Register	Operation specific to Given Error State
Error active	Transmission	0 to 95	TECS1, TECS0 = 00	- Outputs an active error flag (6 consecutive dominant-level bits) on detection of the error.
	Reception	0 to 95	RECS1, RECS0 = 00	
	Transmission	96 to 127	TECS1, TECS0 = 01	
	Reception	96 to 127	RECS1, RECS0 = 01	
Error passive	Transmission	128 to 255	TECS1, TECS0 = 11	- Outputs a passive error flag (6 consecutive recessive-level bits) on detection of the error. - Transmits 8 recessive-level bits, in between transmissions, following an intermission (suspend transmission).
	Reception	128 or more	RECS1, RECS0 = 11	
Bus-off	Transmission	256 or more (not indicated)[Note]	BOFF = 1, TECS1, TECS0 = 11	- Communication is not possible. Messages are not stored when receiving frames, however, the following operations of <1>, <2>, and <3> are done. <1> TSOUT toggles. <2> REC is incremented/decremented. <3> VALID bit is set. - If the CAN module is entered to the initialization mode and then transition request to any operation mode is made, and when 11 consecutive recessive-level bits are detected 128 times, the error counter is reset to 0 and the error active state can be restored.

Note The value of the transmission error counter (TEC) is invalid when the BOFF bit is set to 1. If an error that increments the value of the transmission error counter by +8 while the counter value is in a range of 248 to 255, the counter is not incremented and the bus-off state is assumed.

(b) Error counter

The error counter counts up when an error has occurred, and counts down upon successful transmission and reception. The error counter is updated immediately after error detection.

Table 13-14. Error Counter

State	Transmission Error Counter (TEC7 to TEC0)	Reception Error Counter (REC6 to REC0)
Receiving node detects an error (except bit error in the active error flag or overload flag).	No change	+1 (when REPS bit = 0)
Receiving node detects dominant level following error flag of error frame.	No change	+8 (when REPS bit = 0)
Transmitting node transmits an error flag. [As exceptions, the error counter does not change in the following cases.] <1> ACK error is detected in error passive state and dominant level is not detected while the passive error flag is being output. <2> A stuff error is detected in an arbitration field that transmitted a recessive level as a stuff bit, but a dominant level is detected.	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active transmitting node)	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active receiving node)	No change	+8 (when REPS bit = 0)
When the node detects 14 consecutive dominant-level bits from the beginning of the active error flag or overload flag, and then subsequently detects 8 consecutive dominant-level bits. When the node detects 8 consecutive dominant levels after a passive error flag	+8 (during transmission)	+8 (during reception, when REPS bit = 0)
When the transmitting node has completed transmission without error (±0 if error counter = 0)	–1	No change
<R> When the receiving node has completed reception without error	No change	- –1 (1 ≤ REC6 to REC0 ≤ 127, when REPS bit = 0) - ±0 (REC6 to REC0 = 0, when REPS bit = 0) - Ones of the values from 119 to 126 is set up with the value of a reception error counter of the moment that REPS is set (1). (when REPS bit = 1)

(c) Occurrence of bit error in intermission

An overload frame is generated.

Caution If an error occurs, the error flag output (active or passive) is controlled according to the contents of the transmission error counter and reception error counter before the error occurred. The value of the error counter is incremented after the error flag has been output.

(5) Recovery from bus-off state

When the CAN module is in the bus-off state, the CAN module permanently sets its output signals (CTxD) to recessive level.

The CAN module recovers from the bus-off state in the following bus-off recovery sequence.

<1> A request to enter the CAN initialization mode

<2> A request to enter a CAN operation mode

(a) Recovery operation through normal recovery sequence

(b) Forced recovery operation that skips recovery sequence

(a) Recovery operation from bus-off state through normal recovery sequence

The CAN module first issues a request to enter the initialization mode (see timing <1> in Figure 13-17). This request will be immediately acknowledged, and the OPMODE bits of the CCTRL register are cleared to 000B. Processing such as analyzing the fault that has caused the bus-off state, re-defining the CAN module and message buffer using application software, or stopping the operation of the CAN module can be performed by clearing the GOM bit to 0.

Next, the user requests to change the mode from the initialization mode to an operation mode (see timing <2> in Figure 13-17). This starts an operation to recover the CAN module from the bus-off state. The conditions under which the module can recover from the bus-off state are defined by the CAN protocol ISO 11898, and it is necessary to detect 11 consecutive recessive-level bits 128 times. At this time, the request to change the mode to an operation mode is held pending until the recovery conditions are satisfied. When the recovery conditions are satisfied (see timing <3> in Figure 13-17), the CAN module can enter the operation mode it has requested. Until the CAN module enters this operation mode, it stays in the initialization mode. Completion to be requested operation mode can be confirmed by reading the OPMODE bits of the CCTRL register.

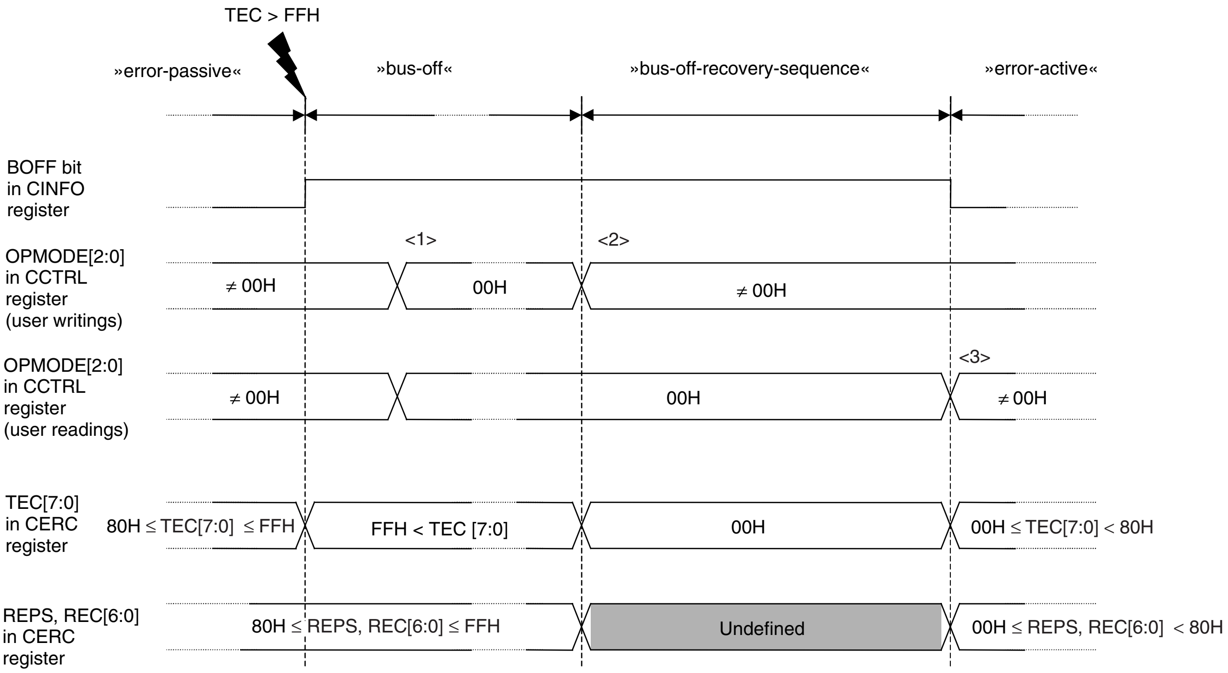
During the bus-off period and bus-off recovery sequence, the BOFF bit of the CINFO register stays set (to 1). In the bus-off recovery sequence, the reception error counter (REC[6:0]) counts the number of times 11 consecutive recessive-level bits have been detected on the bus. Therefore, the recovery state can be checked by reading REC[6:0].

Cautions 1. If the Bus-off Recovery Sequence is interrupted by entering Initialization Mode and re-entering any Operation Mode, the Bus-off Recovery Sequence will restart from the beginning, and the waiting phase will be again 128 times 11 recessive-level bits, counted from this point.

2. In the bus-off recovery sequence, REC[6:0] counts up (+1) each time 11 consecutive recessive-level bits have been detected. Even during the bus-off period, the CAN module can enter the CAN sleep mode or CAN stop mode. To start the bus-off recovery sequence, it is necessary to transit to the initialization mode once.

However, when the CAN module is in either CAN sleep mode or CAN stop mode, transition request to the initialization mode is not accepted, thus you have to release the CAN sleep mode first. In this case, as soon as the CAN sleep mode is released, the bus-off recovery sequence starts and no transition to initialization mode is necessary. If the CAN module detects a dominant edge on the CAN bus while in sleep mode even during bus-off, the sleep mode will be left and the bus-off recovery sequence will start (In the state that the CAN clock is supplied, it is necessary to clear the PSMODE by software after dominant edge detection).

Figure 13-17. Recovery Operation from Bus-off State through Normal Recovery Sequence


TEC > FFH
»error-passive«
»bus-off«
»bus-off-recovery-sequence«
»error-active«
BOFF bit in CINFO register
OPMODE[2:0] in CCTRL register (user writings)
<1>
<2>
≠ 00H
00H
≠ 00H
OPMODE[2:0] in CCTRL register (user readings)
<3>
≠ 00H
00H
≠ 00H
TEC[7:0] in CERC register
80H ≤ TEC[7:0] ≤ FFH
FFH < TEC [7:0]
00H
00H ≤ TEC[7:0] < 80H
REPS, REC[6:0] in CERC register
80H ≤ REPS, REC[6:0] ≤ FFH
Undefined
00H ≤ REPS, REC[6:0] < 80H

(b) Forced recovery operation that skips bus-off recovery sequence

The CAN module can be forcibly released from the bus-off state, regardless of the bus state, by skipping the bus-off recovery sequence. Here is the procedure.

First, the CAN module requests to enter the initialization mode. For the operation and points to be noted at this time, see **(a) Recovery operation from bus-off state through normal recovery sequence**.

Next, the module requests to enter an operation mode. At the same time, the CCERC bit of the CCTRL register must be set to 1.

As a result, the bus-off recovery sequence defined by the CAN protocol ISO 11898 is skipped, and the module immediately enters the operation mode. In this case, the module is connected to the CAN bus after it has monitored 11 consecutive recessive-level bits. For details, see the processing in Figure 13-82.

Caution This function is not defined by the CAN protocol ISO 11898. When using this function, thoroughly evaluate its effect on the network system.

(6) Initializing CAN module error counter register (CERC) in initialization mode

If it is necessary to initialize the CAN module error counter register (CERC) and CAN module information register (CINFO) for debugging or evaluating a program, they can be initialized to the default value by setting the CCERC bit of the CCTRL register in the initialization mode. When initialization has been completed, the CCERC bit is automatically cleared to 0.

Cautions 1. This function is enabled only in the initialization mode. Even if the CCERC bit is set to 1 in a CAN operation mode, the CERC and CINFO registers are not initialized.

2. The CCERC bit can be set at the same time as the request to enter a CAN operation mode.

13.3.7 Baud rate control function

(1) Prescaler

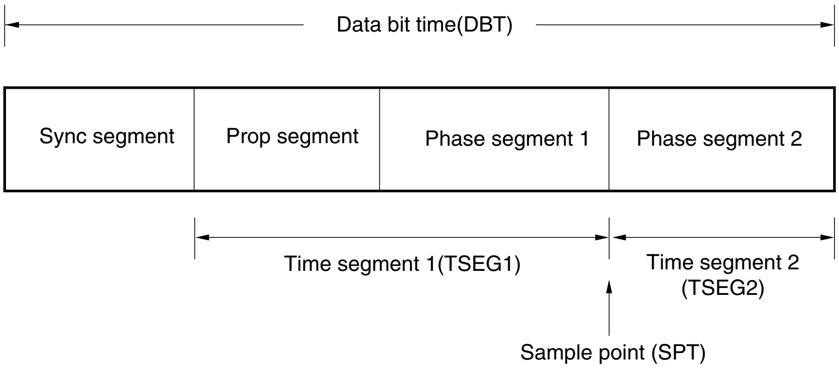
The CAN controller has a prescaler that divides the clock (f_{CAN}) supplied to CAN. This prescaler generates a CAN protocol layer basic clock (f_{TQ}) derived from the CAN module system clock (f_{CANMOD}), and divided by 1 to 256 (see **13.6 (12) CAN Bit Rate Prescaler Register (CBRP)**).

(2) Data bit time (8-25 time quanta)

One data bit time is defined as shown in Figure 13-18.

The CAN controller sets time segment 1, time segment 2, and reSynchronization Jump Width (SJW) as the parameter of data bit time, as shown in Figure 13-18. Time segment 1 is equivalent to the total of the propagation (prop) segment and phase segment 1 that are defined by the CAN protocol specification. Time segment 2 is equivalent to phase segment 2.

Figure 13-18. Segment Setting



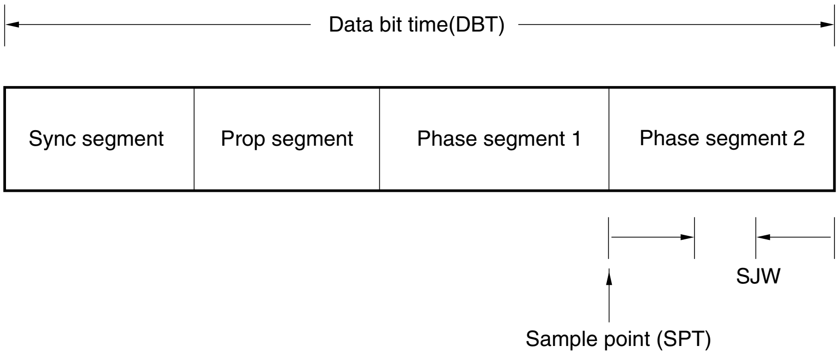
<R>

Segment Name	Settable Range	Notes on Setting to Confirm to CAN Specification
Time Segment 1 (TSEG1)	2TQ-16TQ	The length which added 1TQ to the length of the time segment 2, however a minimum of 2 TQ(s) serve as a setting minimum of the time segment 1.
Time Segment 2 (TSEG2)	1TQ-8TQ	IPT of the CAN controller is 0TQ. To conform to the CAN protocol specification, the length below the phase segment 1 must be set up here. This means that the length of the time segment 1 minus 1TQ is the settable upper limit of the time segment 2. (Max. 8TQ)
Resynchronization jump width(SJW)	1TQ-4TQ	The length equal to the time segment 2, or which of 4TQ or the length of the smaller one serves as a setting maximum of synchronous jump width.

Remark IPT : Information Processing Time
TQ : Time Quanta

Reference: The CAN standard ISO 11898 specification defines the segments constituting the data bit time as shown in Figure 13-19.

Figure 13-19. Reference: Configuration of Data Bit Time Defined by CAN Specification



Segment Name	Segment Length	Description
Sync Segment (Synchronization Segment)	1	This segment starts at the edge where the level changes from recessive to dominant when hard-synchronization is established.
Prop Segment	Programmable to 1 to 8 or more	This segment absorbs the delay of the output buffer, CAN bus, and input buffer. The length of this segment is set so that ACK is returned before the start of phase segment 1. Time of prop segment ≥ (Delay of output buffer) + 2 × (Delay of CAN bus) + (Delay of input buffer)
Phase Segment 1	Programmable to 1 to 8	This segment compensates for an error of data bit time. The longer this segment, the wider the permissible range but the slower the communication speed.
Phase Segment 2	Phase Segment 1 or IPT, whichever greater	
SJW	Programmable from 1TQ to length of segment 1 or 4TQ, whichever is smaller	This width sets the upper limit of expansion or contraction of the phase segment during resynchronization.

Remark IPT : Information Processing Time
TQ : Time Quanta

(3) Synchronizing data bit

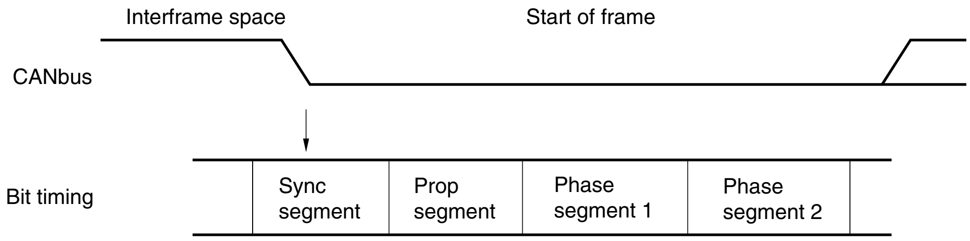
- The receiving node establishes synchronization by a level change on the bus because it does not have a sync signal.
- The transmitting node transmits data in synchronization with the bit timing of the transmitting node.

(a) Hard-synchronization

This synchronization is established when the receiving node detects the start of frame in the interframe space.

- When a falling edge is detected on the bus, that TQ means the sync segment and the next segment is the prop segment. In this case, synchronization is established regardless of SJW.

Figure 13-20. Hard-synchronization at Recognition of Dominant Level during Bus Idle

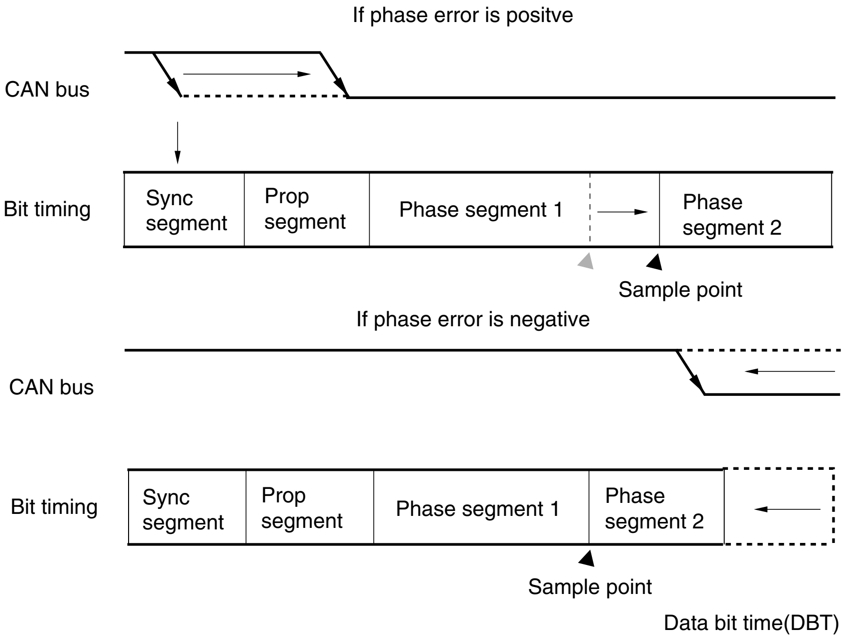


(b) Resynchronization

Synchronization is established again if a level change is detected on the bus during reception (only if a recessive level was sampled previously).

- The phase error of the edge is given by the relative position of the detected edge and sync segment.
  <Sign of phase error>
    0: If the edge is within the sync segment
    Positive: If the edge is before the sample point (phase error)
    Negative: If the edge is after the sample point (phase error)
    If phase error is positive: Phase segment 1 is longer by specified SJW.
    If phase error is negative: Phase segment 2 is shorter by specified SJW.
- The sample point of the data of the receiving node moves relatively due to the “discrepancy” in baud rate between the transmitting node and receiving node.

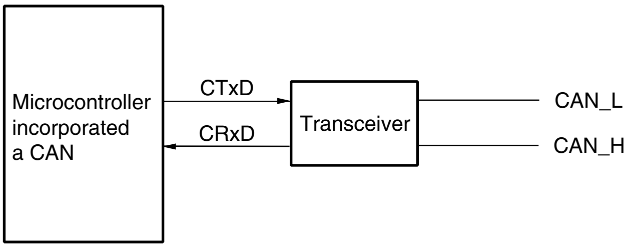
Figure 13-21. Resynchronization



13.4 Connection with Target System

The microcontroller incorporated a CAN has to be connected to the CAN bus using an external transceiver.

Figure 13-22. Connection to CAN Bus



13.5 Internal Registers of CAN Controller

13.5.1 CAN controller configuration

Table 13-15. List of CAN Controller Registers (1/2)

Item	Register Name
Control registers	Peripheral clock select register (PCKSEL)
	Serial communication pin select register (STSEL)
	Port register 1, 7 (P1, P7)
	Port mode register 1, 7 (PM1, PM7)
CAN global registers	CAN global module control register (CGMCTRL)
	CAN global module clock select register (CGMCS)
	CAN global automatic block transmission control register (CGMABT)
	CAN global automatic block transmission delay setting register (CGMABTD)
CAN module registers	CAN module mask 1 register (CMASK1L, CMASK1H)
	CAN module mask 2 register (CMASK2L, CMASK2H)
	CAN module mask 3 register (CMASK3L, CMASK3H)
	CAN module mask 4 register (CMASK4L, CMASK4H)
	CAN module control register (CCTRL)
	CAN module last error code register (CLEC)
	CAN module information register (CINFO)
	CAN module error counter register (CERC)
	CAN module interrupt enable register (CIE)
	CAN module interrupt status register (CINTS)
	CAN module bit rate prescaler register (CBRP)
	CAN module bit rate register (CBTR)
	CAN module last in-pointer register (CLIPT)
	CAN module receive history list register (CRGPT)
	CAN module last out-pointer register (CLOPT)
	CAN module transmit history list register (CTGPT)
	CAN module time stamp register (CTS)

Remarks 1. CAN global registers are identified by CGM<register function>.
CAN module registers are identified by C<register function>.
Message buffer registers are identified by CM<register function>.

2. m = 0 to 15

Table 13-15. List of CAN Controller Registers (2/2)

Item	Register Name
Message buffer registers	CAN message data byte 01 register m (CMDB01m)
	CAN message data byte 0 register m (CMDB0m)
	CAN message data byte 1 register m (CMDB1m)
	CAN message data byte 23 register m (CMDB23m)
	CAN message data byte 2 register m (CMDB2m)
	CAN message data byte 3 Register m (CMDB3m)
	CAN message data byte 45 Register m (CMDB45m)
	CAN message data byte 4 Register m (CMDB4m)
	CAN message data byte 5 Register m (CMDB5m)
	CAN message data byte 67 Register m (CMDB67m)
	CAN message data byte 6 register m (CMDB6m)
	CAN message data byte 7 register m (CMDB7m)
	CAN message data length register m (CMDLCm)
	CAN message configuration register m (CMCONFm)
	CAN message ID register m (CMIDLm, CMIDHm)
	CAN message control register m (CMCTRLm)

Remarks 1. CAN global registers are identified by CGM<register function>.
CAN module registers are identified by C<register function>.
Message buffer registers are identified by CM<register function>.

2. m = 0 to 15

13.5.2 Register access type

The peripheral I/O register for the CAN controller is assigned to 000F05C0H to 000F06FFH. For details, see **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 13-16. Register Access Types (1/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F05C0H	CAN global module control register	CGMCTRL	R/W	–	–	√	0000H
000F05C6H	CAN global automatic block transmission control register	CGMABT	R/W	–	–	√	0000H
000F05C8H	CAN global automatic block transmission delay setting register	CGMABTD	R/W	–	√	–	00H
000F05CEH	CAN global module clock select register	CGMCS	R/W	–	√	–	0FH
000F05D0H	CAN module mask 1 register	CMASK1L	R/W	–	–	√	Undefined
000F05D2H		CMASK1H					Undefined
000F05D4H	CAN module mask 2 register	CMASK2L	R/W	–	–	√	Undefined
000F05D6H		CMASK2H					Undefined
000F05D8H	CAN module mask 3 register	CMASK3L	R/W	–	–	√	Undefined
000F05DAH		CMASK3H					Undefined
000F05DCH	CAN module mask 4 register	CMASK4L	R/W	–	–	√	Undefined
000F05DEH		CMASK4H					Undefined
000F05E0H	CAN module control register	CCTRL	R/W	–	–	√	0000H
000F05E2H	CAN module last error code register	CLEC	R/W	–	√	–	00H
000F05E3H	CAN module information register	CINFO	R	–	√	–	00H
000F05E4H	CAN module error counter register	CERC	R	–	–	√	0000H
000F05E6H	CAN module interrupt enable register	CIE	R/W	–	–	√	0000H
000F05E8H	CAN module interrupt status register	CINTS	R/W	–	–	√	0000H
000F05EAH	CAN module bit rate prescaler register	CBRP	R/W	–	√	–	FFH
000F05ECH	CAN module bit rate register	CBTR	R/W	–	–	√	370FH
000F05EEH	CAN module last in-pointer register	CLIPT	R	–	√	–	Undefined
000F05F0H	CAN module receive history list register	CRGPT	R/W	–	–	√	xx02H
000F05F2H	CAN module last out-pointer register	CLOPT	R	–	√	–	Undefined
000F05F4H	CAN module transmit history list register	CTGPT	R/W	–	–	√	xx02H
000F05F6H	CAN module time stamp register	CTS	R/W	–	–	√	0000H

Table 13-16. Register Access Types (2/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0600H	CAN message data byte 01 register 00	CMDB0100	R/W	–	–	√	Undefined
000F0600H	CAN message data byte 0 register 00	CMDB000		–	√	–	Undefined
000F0601H	CAN message data byte 1 register 00	CMDB100		–	√	–	Undefined
000F0602H	CAN message data byte 23 register 00	CMDB2300		–	–	√	Undefined
000F0602H	CAN message data byte 2 register 00	CMDB200		–	√	–	Undefined
000F0603H	CAN message data byte 3 register 00	CMDB300		–	√	–	Undefined
000F0604H	CAN message data byte 45 register 00	CMDB4500		–	–	√	Undefined
000F0604H	CAN message data byte 4 register 00	CMDB400		–	√	–	Undefined
000F0605H	CAN message data byte 5 register 00	CMDB500		–	√	–	Undefined
000F0606H	CAN message data byte 67 register 00	CMDB6700		–	–	√	Undefined
000F0606H	CAN message data byte 6 register 00	CMDB600		–	√	–	Undefined
000F0607H	CAN message data byte 7 register 00	CMDB700		–	√	–	Undefined
000F0608H	CAN message data length register 00	CMDLC00		–	√	–	0000xxxxB
000F0609H	CAN message configuration register 00	CMCONF00		–	√	–	Undefined
000F060AH	CAN message ID register 00	CMIDL00		–	–	√	Undefined
000F060CH		CMIDH00		–	–	√	Undefined
000F060EH	CAN message control register 00	CMCTRL00		–	–	√	00x00000 000xx000B
000F0610H	CAN message data byte 01 register 01	CMDB0101		–	–	√	Undefined
000F0610H	CAN message data byte 0 register 01	CMDB001		–	√	–	Undefined
000F0611H	CAN message data byte 1 register 01	CMDB101		–	√	–	Undefined
000F0612H	CAN message data byte 23 register 01	CMDB2301		–	–	√	Undefined
000F0612H	CAN message data byte 2 register 01	CMDB201		–	√	–	Undefined
000F0613H	CAN message data byte 3 register 01	CMDB301		–	√	–	Undefined
000F0614H	CAN message data byte 45 register 01	CMDB4501		–	–	√	Undefined
000F0614H	CAN message data byte 4 register 01	CMDB401		–	√	–	Undefined
000F0615H	CAN message data byte 5 register 01	CMDB501		–	√	–	Undefined
000F0616H	CAN message data byte 67 register 01	CMDB6701		–	–	√	Undefined
000F0616H	CAN message data byte 6 register 01	CMDB601		–	√	–	Undefined
000F0617H	CAN message data byte 7 register 01	CMDB701		–	√	–	Undefined
000F0618H	CAN message data length register 01	CMDLC01		–	√	–	0000xxxxB
000F0619H	CAN message configuration register 01	CMCONF01		–	√	–	Undefined
000F061AH	CAN message ID register 01	CMIDL01		–	–	√	Undefined
000F061CH		CMIDH01		–	–	√	Undefined
000F061EH	CAN message control register 01	CMCTRL01		–	–	√	00x00000 000xx000B

Table 13-16. Register Access Types (3/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0620H	CAN message data byte 01 register 02	CMDB0102	R/W	–	–	√	Undefined
000F0620H	CAN message data byte 0 register 02	CMDB002		–	√	–	Undefined
000F0621H	CAN message data byte 1 register 02	CMDB102		–	√	–	Undefined
000F0622H	CAN message data byte 23 register 02	CMDB2302		–	–	√	Undefined
000F0622H	CAN message data byte 2 register 02	CMDB202		–	√	–	Undefined
000F0623H	CAN message data byte 3 register 02	CMDB302		–	√	–	Undefined
000F0624H	CAN message data byte 45 register 02	CMDB4502		–	–	√	Undefined
000F0624H	CAN message data byte 4 register 02	CMDB402		–	√	–	Undefined
000F625H	CAN message data byte 5 register 02	CMDB502		–	√	–	Undefined
000F0626H	CAN message data byte 67 register 02	CMDB6702		–	–	√	Undefined
000F0626H	CAN message data byte 6 register 02	CMDB602		–	√	–	Undefined
000F0627H	CAN message data byte 7 register 02	CMDB702		–	√	–	Undefined
000F0628H	CAN message data length register 02	CMDLC02		–	√	–	0000xxxxB
000F0629H	CAN message configuration register 02	CMCONF02		–	√	–	Undefined
000F062AH	CAN message ID register 02	CMIDL02		–	–	√	Undefined
000F062CH		CMIDH02		–	–	√	Undefined
000F062EH	CAN message control register 02	CMCTRL02		–	–	√	00x00000 000xx000B
000F0630H	CAN message data byte 01 register 03	CMDB0103		–	–	√	Undefined
000F0630H	CAN message data byte 0 register 03	CMDB003		–	√	–	Undefined
000F0631H	CAN message data byte 1 register 03	CMDB103		–	√	–	Undefined
000F0632H	CAN message data byte 23 register 03	CMDB2303		–	–	√	Undefined
000F0632H	CAN message data byte 2 register 03	CMDB203		–	√	–	Undefined
000F0633H	CAN message data byte 3 register 03	CMDB303		–	√	–	Undefined
000F0634H	CAN message data byte 45 register 03	CMDB4503		–	–	√	Undefined
000F0634H	CAN message data byte 4 register 03	CMDB403		–	√	–	Undefined
000F0635H	CAN message data byte 5 register 03	CMDB503		–	√	–	Undefined
000F0636H	CAN message data byte 67 register 03	CMDB6703		–	–	√	Undefined
000F0636H	CAN message data byte 6 register 03	CMDB603		–	√	–	Undefined
000F0637H	CAN message data byte 7 register 03	CMDB703		–	√	–	Undefined
000F0638H	CAN message data length register 03	CMDLC03		–	√	–	0000xxxxB
000F0639H	CAN message configuration register 03	CMCONF03		–	√	–	Undefined
000F063AH	CAN message ID register 03	CMIDL03		–	–	√	Undefined
000F063CH		CMIDH03		–	–	√	Undefined
000F063EH	CAN message control register 03	CMCTRL03		–	–	√	00x00000 000xx000B

Table 13-16. Register Access Types (4/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0640H	CAN message data byte 01 register 04	CMDB0104	R/W	–	–	√	Undefined
000F0640H	CAN message data byte 0 register 04	CMDB004		–	√	–	Undefined
000F0641H	CAN message data byte 1 register 04	CMDB104		–	√	–	Undefined
000F0642H	CAN message data byte 23 register 04	CMDB2304		–	–	√	Undefined
000F0642H	CAN message data byte 2 register 04	CMDB204		–	√	–	Undefined
000F0643H	CAN message data byte 3 register 04	CMDB304		–	√	–	Undefined
000F0644H	CAN message data byte 45 register 04	CMDB4504		–	–	√	Undefined
000F0644H	CAN message data byte 4 register 04	CMDB404		–	√	–	Undefined
000F0645H	CAN message data byte 5 register 04	CMDB504		–	√	–	Undefined
000F0646H	CAN message data byte 67 register 04	CMDB6704		–	–	√	Undefined
000F0646H	CAN message data byte 6 register 04	CMDB604		–	√	–	Undefined
000F0647H	CAN message data byte 7 register 04	CMDB704		–	√	–	Undefined
000F0648H	CAN message data length register 04	CMDLC04		–	√	–	0000xxxxB
000F0649H	CAN message configuration register 04	CMCONF04		–	√	–	Undefined
000F064AH	CAN message ID register 04	CMIDL04		–	–	√	Undefined
000F064CH		CMIDH04		–	–	√	Undefined
000F064EH	CAN message control register 04	CMCTRL04		–	–	√	00x00000 000xx000B
000F0650H	CAN message data byte 01 register 05	CMDB0105		–	–	√	Undefined
000F0650H	CAN message data byte 0 register 05	CMDB005		–	√	–	Undefined
000F0651H	CAN message data byte 1 register 05	CMDB105		–	√	–	Undefined
000F0652H	CAN message data byte 23 register 05	CMDB2305		–	–	√	Undefined
000F0652H	CAN message data byte 2 register 05	CMDB205		–	√	–	Undefined
000F0653H	CAN message data byte 3 register 05	CMDB305		–	√	–	Undefined
000F0654H	CAN message data byte 45 register 05	CMDB4505		–	–	√	Undefined
000F0654H	CAN message data byte 4 register 05	CMDB405		–	√	–	Undefined
000F0655H	CAN message data byte 5 register 05	CMDB505		–	√	–	Undefined
000F0656H	CAN message data byte 67 register 05	CMDB6705		–	–	√	Undefined
000F0656H	CAN message data byte 6 register 05	CMDB605		–	√	–	Undefined
000F0657H	CAN message data byte 7 register 05	CMDB705		–	√	–	Undefined
000F0658H	CAN message data length register 05	CMDLC05		–	√	–	0000xxxxB
000F0659H	CAN message configuration register 05	CMCONF05		–	√	–	Undefined
000F065AH	CAN message ID register 05	CMIDL05		–	–	√	Undefined
000F065CH		CMIDH05		–	–	√	Undefined
000F065EH	CAN message configuration register 05	CMCTRL05		–	–	√	00x00000 000xx000B

Table 13-16. Register Access Types (5/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0660H	CAN message data byte 01 register 06	CMDB0106	R/W	–	–	√	Undefined
000F0660H	CAN message data byte 0 register 06	CMDB006		–	√	–	Undefined
000F0661H	CAN message data byte 1 register 06	CMDB106		–	√	–	Undefined
000F0662H	CAN message data byte 23 register 06	CMDB2306		–	–	√	Undefined
000F0662H	CAN message data byte 2 register 06	CMDB206		–	√	–	Undefined
000F0663H	CAN message data byte 3 register 06	CMDB306		–	√	–	Undefined
000F0664H	CAN message data byte 45 register 06	CMDB4506		–	–	√	Undefined
000F0664H	CAN message data byte 4 register 06	CMDB406		–	√	–	Undefined
000F0665H	CAN message data byte 5 register 06	CMDB506		–	√	–	Undefined
000F0666H	CAN message data byte 67 register 06	CMDB6706		–	–	√	Undefined
000F0666H	CAN message data byte 6 register 06	CMDB606		–	√	–	Undefined
000F0667H	CAN message data byte 7 register 06	CMDB706		–	√	–	Undefined
000F0668H	CAN message data length register 06	CMDLC06		–	√	–	0000xxxxB
000F0669H	CAN message configuration register 06	CMCONF06		–	√	–	Undefined
000F066AH	CAN message ID register 06	CMIDL06		–	–	√	Undefined
000F066CH		CMIDH06		–	–	√	Undefined
000F066EH	CAN message control register 06	CMCTRL06		–	–	√	00x00000 000xx000B
000F0670H	CAN message data byte 01 register 07	CMDB0107		–	–	√	Undefined
000F0670H	CAN message data byte 0 register 07	CMDB007		–	√	–	Undefined
000F0671H	CAN message data byte 1 register 07	CMDB107		–	√	–	Undefined
000F0672H	CAN message data byte 23 register 07	CMDB2307		–	–	√	Undefined
000F0672H	CAN message data byte 2 register 07	CMDB207		–	√	–	Undefined
000F0673H	CAN message data byte 3 register 07	CMDB307		–	√	–	Undefined
000F0674H	CAN message data byte 45 register 07	CMDB4507		–	–	√	Undefined
000F0674H	CAN message data byte 4 register 07	CMDB407		–	√	–	Undefined
000F0675H	CAN message data byte 5 register 07	CMDB507		–	√	–	Undefined
000F0676H	CAN message data byte 67 register 07	CMDB6707		–	–	√	Undefined
000F0676H	CAN message data byte 6 register 07	CMDB607		–	√	–	Undefined
000F0677H	CAN message data byte 7 register 07	CMDB707		–	√	–	Undefined
000F0678H	CAN message data length register 07	CMDLC07		–	√	–	0000xxxxB
000F0679H	CAN message configuration register 07	CMCONF07		–	√	–	Undefined
000F067AH	CAN message ID register 07	CMIDL07		–	–	√	Undefined
000F067CH		CMIDH07		–	–	√	Undefined
000F067EH	CAN message control register 07	CMCTRL07		–	–	√	00x00000 000xx000B

Table 13-16. Register Access Types (6/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0680H	CAN message data byte 01 register 08	CMDB0108	R/W	–	–	√	Undefined
000F0680H	CAN message data byte 0 register 08	CMDB008		–	√	–	Undefined
000F0681H	CAN message data byte 1 register 08	CMDB108		–	√	–	Undefined
000F0682H	CAN message data byte 23 register 08	CMDB2308		–	–	√	Undefined
000F0682H	CAN message data byte 2 register 08	CMDB208		–	√	–	Undefined
000F0683H	CAN message data byte 3 register 08	CMDB308		–	√	–	Undefined
000F0684H	CAN message data byte 45 register 08	CMDB4508		–	–	√	Undefined
000F0684H	CAN message data byte 4 register 08	CMDB408		–	√	–	Undefined
000F0685H	CAN message data byte 5 register 08	CMDB508		–	√	–	Undefined
000F0686H	CAN message data byte 67 register 08	CMDB6708		–	–	√	Undefined
000F0686H	CAN message data byte 6 register 08	CMDB608		–	√	–	Undefined
000F0687H	CAN message data byte 7 register 08	CMDB708		–	√	–	Undefined
000F0688H	CAN message data length register 08	CMDLC08		–	√	–	0000xxxxB
000F0689H	CAN message configuration register 08	CMCONF08		–	√	–	Undefined
000F068AH	CAN message ID register 08	CMIDL08		–	–	√	Undefined
000F068CH		CMIDH08		–	–	√	Undefined
000F068EH	CAN message control register 08	CMCTRL08		–	–	√	00x00000000xx000B
000F0690H	CAN message data byte 01 register 09	CMDB0109		–	–	√	Undefined
000F0690H	CAN message data byte 0 register 09	CMDB009		–	√	–	Undefined
000F0691H	CAN message data byte 1 register 09	CMDB109		–	√	–	Undefined
000F0692H	CAN message data byte 23 register 09	CMDB2309		–	–	√	Undefined
000F0692H	CAN message data byte 2 register 09	CMDB209		–	√	–	Undefined
000F0693H	CAN message data byte 3 register 09	CMDB309		–	√	–	Undefined
000F0694H	CAN message data byte 45 register 09	CMDB4509		–	–	√	Undefined
000F0694H	CAN message data byte 4 register 09	CMDB409		–	√	–	Undefined
000F0695H	CAN message data byte 5 register 09	CMDB509		–	√	–	Undefined
000F0696H	CAN message data byte 67 register 09	CMDB6709		–	–	√	Undefined
000F0696H	CAN message data byte 6 register 09	CMDB609		–	√	–	Undefined
000F0697H	CAN message data byte 7 register 09	CMDB709		–	√	–	Undefined
000F0698H	CAN message data length register 09	CMDLC09		–	√	–	0000xxxxB
000F0699H	CAN message configuration register 09	CMCONF09		–	√	–	Undefined
000F069AH	CAN message ID register 09	CMIDL09		–	–	√	Undefined
000F069CH		CMIDH09		–	–	√	Undefined
000F069EH	CAN message control register 09	CMCTRL09		–	–	√	00x00000000xx000B

Table 13-16. Register Access Types (7/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F06A0H	CAN message data byte 01 register 10	CMDB0110	R/W	–	–	√	Undefined
000F06A0H	CAN message data byte 0 register 10	CMDB010		–	√	–	Undefined
000F06A1H	CAN message data byte 1 register 10	CMDB110		–	√	–	Undefined
000F06A2H	CAN message data byte 23 register 10	CMDB2310		–	–	√	Undefined
000F06A2H	CAN message data byte 2 register 10	CMDB210		–	√	–	Undefined
000F06A3H	CAN message data byte 3 register 10	CMDB310		–	√	–	Undefined
000F06A4H	CAN message data byte 45 register 10	CMDB4510		–	–	√	Undefined
000F06A4H	CAN message data byte 4 register 10	CMDB410		–	√	–	Undefined
000F06A5H	CAN message data byte 5 register 10	CMDB510		–	√	–	Undefined
000F06A6H	CAN message data byte 67 register 10	CMDB6710		–	–	√	Undefined
000F06A6H	CAN message data byte 6 register 10	CMDB610		–	√	–	Undefined
000F06A7H	CAN message data byte 7 register 10	CMDB710		–	√	–	Undefined
000F06A8H	CAN message data length register 10	CMDLC10		–	√	–	0000xxxxB
000F06A9H	CAN message configuration register 10	CMCONF10		–	√	–	Undefined
000F06AAH	CAN message ID register 10	CMIDL10		–	–	√	Undefined
000F06ACH		CMIDH10		–	–	√	Undefined
000F06AEH	CAN message control register 10	CMCTRL10		–	–	√	00x00000 000xx000B
000F06B0H	CAN message data byte 01 register 11	CMDB0111		–	–	√	Undefined
000F06B0H	CAN message data byte 0 register 11	CMDB011		–	√	–	Undefined
000F06B1H	CAN message data byte 1 register 11	CMDB111		–	√	–	Undefined
000F06B2H	CAN message data byte 23 register 11	CMDB2311		–	–	√	Undefined
000F06B2H	CAN message data byte 2 register 11	CMDB211		–	√	–	Undefined
000F06B3H	CAN message data byte 3 register 11	CMDB311		–	√	–	Undefined
000F06B4H	CAN message data byte 45 register 11	CMDB4511		–	–	√	Undefined
000F06B4H	CAN message data byte 4 register 11	CMDB411		–	√	–	Undefined
000F06B5H	CAN message data byte 51 register 11	CMDB511		–	√	–	Undefined
000F06B6H	CAN message data byte 67 register 11	CMDB6711		–	–	√	Undefined
000F06B6H	CAN message data byte 6 register 11	CMDB611		–	√	–	Undefined
000F06B7H	CAN message data byte 71 register 11	CMDB711		–	√	–	Undefined
000F06B8H	CAN message data length register 11	CMDLC11		–	√	–	0000xxxxB
000F06B9H	CAN message configuration register 11	CMCONF11		–	√	–	Undefined
000F06BAH	CAN message ID register 11	CMIDL11		–	–	√	Undefined
000F06BCH		CMIDH11		–	–	√	Undefined
000F06BEH	CAN message control register 11	CMCTRL11		–	–	√	00x00000 000xx000B

Table 13-16. Register Access Types (8/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F06C0H	CAN message data byte 01 register 12	CMDB0112	R/W	–	–	√	Undefined
000F06C0H	CAN message data byte 0 register 12	CMDB012		–	√	–	Undefined
000F06C1H	CAN message data byte 1 register 12	CMDB112		–	√	–	Undefined
000F06C2H	CAN message data byte 23 register 12	CMDB2312		–	–	√	Undefined
000F06C2H	CAN message data byte 2 register 12	CMDB212		–	√	–	Undefined
000F06C3H	CAN message data byte 3 register 12	CMDB312		–	√	–	Undefined
000F06C4H	CAN message data byte 45 register 12	CMDB4512		–	–	√	Undefined
000F06C4H	CAN message data byte 4 register 12	CMDB412		–	√	–	Undefined
000F06C5H	CAN message data byte 5 register 12	CMDB512		–	√	–	Undefined
000F06C6H	CAN message data byte 67 register 12	CMDB6712		–	–	√	Undefined
000F06C6H	CAN message data byte 6 register 12	CMDB612		–	√	–	Undefined
000F06C7H	CAN message data byte 7 register 12	CMDB712		–	√	–	Undefined
000F06C8H	CAN message data length register 12	CMDLC12		–	√	–	0000xxxxB
000F06C9H	CAN message configuration register 12	CMCONF12		–	√	–	Undefined
000F06CAH	CAN message ID register 12	CMIDL12		–	–	√	Undefined
000F06CCH		CMIDH12		–	–	√	Undefined
000F06CEH	CAN message control register 12	CMCTRL12		–	–	√	00x00000 000xx000B
000F06D0H	CAN message data byte 01 register 13	CMDB0113		–	–	√	Undefined
000F06D0H	CAN message data byte 0 register 13	CMDB013		–	√	–	Undefined
000F06D1H	CAN message data byte 1 register 13	CMDB113		–	√	–	Undefined
000F06D2H	CAN message data byte 23 register 13	CMDB2313		–	–	√	Undefined
000F06D2H	CAN message data byte 2 register 13	CMDB213		–	√	–	Undefined
000F06D3H	CAN message data byte 3 register 13	CMDB313		–	√	–	Undefined
000F06D4H	CAN message data byte 45 register 13	CMDB4513		–	–	√	Undefined
000F06D4H	CAN message data byte 4 register 13	CMDB413		–	√	–	Undefined
000F06D5H	CAN message data byte 5 register 13	CMDB513		–	√	–	Undefined
000F06D6H	CAN message data byte 67 register 13	CMDB6713		–	–	√	Undefined
000F06D6H	CAN message data byte 6 register 13	CMDB613		–	√	–	Undefined
000F06D7H	CAN message data byte 7 register 13	CMDB713		–	√	–	Undefined
000F06D8H	CAN message data length register 13	CMDLC13		–	√	–	0000xxxxB
000F06D9H	CAN message configuration register 13	CMCONF13		–	√	–	Undefined
000F06DAH	CAN message ID register 13	CMIDL13		–	–	√	Undefined
000F06DCH		CMIDH13		–	–	√	Undefined
000F06DEH	CAN message control register 13	CMCTRL13		–	–	√	00x00000 000xx000B

Table 13-16. Register Access Types (9/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F06E0H	CAN message data byte 01 register 14	CMDB0114	R/W	–	–	√	Undefined
000F06E0H	CAN message data byte 0 register 14	CMDB014		–	√	–	Undefined
000F06E1H	CAN message data byte 1 register 14	CMDB114		–	√	–	Undefined
000F06E2H	CAN message data byte 23 register 14	CMDB2314		–	–	√	Undefined
000F06E2H	CAN message data byte 2 register 14	CMDB214		–	√	–	Undefined
000F06E3H	CAN message data byte 3 register 14	CMDB314		–	√	–	Undefined
000F06E4H	CAN message data byte 45 register 14	CMDB4514		–	–	√	Undefined
000F06E4H	CAN message data byte 4 register 14	CMDB414		–	√	–	Undefined
000F06E5H	CAN message data byte 5 register 14	CMDB514		–	√	–	Undefined
000F06E6H	CAN message data byte 67 register 14	CMDB6714		–	–	√	Undefined
000F06E6H	CAN message data byte 6 register 14	CMDB614		–	√	–	Undefined
000F06E7H	CAN message data byte 7 register 14	CMDB714		–	√	–	Undefined
000F06E8H	CAN message data length register 14	CMDLC14		–	√	–	0000xxxxB
000F06E9H	CAN message configuration register 14	CMCONF14		–	√	–	Undefined
000F06EAH	CAN message ID register 14	CMIDL14		–	–	√	Undefined
000F06ECH		CMIDH14		–	–	√	Undefined
000F06EEH	CAN message control register 14	CMCTRL14		–	–	√	00x00000 000xx000B
000F06F0H	CAN message data byte 01 register 15	CMDB0115		–	–	√	Undefined
000F06F0H	CAN message data byte 0 register 15	CMDB015		–	√	–	Undefined
000F06F1H	CAN message data byte 1 register 15	CMDB115		–	√	–	Undefined
000F06F2H	CAN message data byte 23 register 15	CMDB2315		–	–	√	Undefined
000F06F2H	CAN message data byte 2 register 15	CMDB215		–	√	–	Undefined
000F06F3H	CAN message data byte 3 register 15	CMDB315		–	√	–	Undefined
000F06F4H	CAN message data byte 45 register 15	CMDB4515		–	–	√	Undefined
000F06F4H	CAN message data byte 4 register 15	CMDB415			√	–	Undefined
000F06F5H	CAN message data byte 5 register 15	CMDB515		–	√	–	Undefined
000F06F6H	CAN message data byte 67 register 15	CMDB6715		–	–	√	Undefined
000F06F6H	CAN message data byte 6 register 15	CMDB615		–	√	–	Undefined
000F06F7H	CAN message data byte 7 register 15	CMDB715		–	√	–	Undefined
000F06F8H	CAN message data length register 15	CMDLC15		–	√	–	0000xxxxB
000F06F9H	CAN message configuration register 15	CMCONF15		–	√	–	Undefined
000F06FAH	CAN message ID register 15	CMIDL15		–	–	√	Undefined
000F06FCH		CMIDH15		–	–	√	Undefined
000F06FEH	CAN message control register 15	CMCTRL15		–	–	√	00x00000 000xx000B

13.5.3 Register bit configuration

Table 13-17. Bit Configuration of CAN Global Registers

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
000F05C0H	CGMCTRL(W)	0	0	0	0	0	0	0	Clear GOM
000F05C1H		0	0	0	0	0	0	Set EFSD	Set GOM
000F05C0H	CGMCTRL(R)	0	0	0	0	0	0	EFSD	GOM
000F05C1H		MBON	0	0	0	0	0	0	0
000F05C6H	CGMABT(W)	0	0	0	0	0	0	0	Clear ABTTRG
000F05C7H		0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
000F05C6H	CGMABT(R)	0	0	0	0	0	0	ABTCLR	ABTTRG
000F05C7H		0	0	0	0	0	0	0	0
000F05C8H	CGMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0
000F05CEH	CGMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0

Caution The actual register address is calculated as follows:
Register Address = Global Register Area Offset (CH dependent) + Offset Address as listed in table above

Remark (R) When read
(W) When write

Table 13-18. Bit Configuration of CAN Module Registers (1/2)

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
000F05D0H	CMASK1L	CM1ID [7:0]							
000F05D1H		CM1ID [15:8]							
000F05D2H	CMASK1H	CM1ID [23:16]							
000F05D3H		0	0	0	CM1ID [28:24]				
000F05D4H	CMASK2L	CM2ID [7:0]							
000F05D5H		CM2ID [15:8]							
000F05D6H	CMASK2H	CM2ID [23:16]							
000F05D7H		0	0	0	CM2ID [28:24]				
000F05D8H	CMASK3L	CM3ID [7:0]							
000F05D9H		CM3ID [15:8]							
000F05DAH	CMASK3H	CM3ID [23:16]							
000F05DBH		0	0	0	CM3ID [28:24]				
000F05DCH	CMASK4L	CM4ID [7:0]							
000F05DDH		CM4ID [15:8]							
000F05DEH	CMASK4H	CM4ID [23:16]							
000F05DFH		0	0	0	CM4ID [28:24]				
000F05E0H	CCTRL(W)	Clear CCERC	Clear AL	Clear VALID	Clear PSMODE 1	Clear PSMODE 0	Clear OPMODE 2	Clear OPMODE 1	Clear OPMODE 0
000F05E1H		Set CCERC	Set AL	0	Set PSMODE 1	Set PSMODE 0	Set OPMODE 2	Set OPMODE 1	Set OPMODE 0
000F05E0H	CCTRL(R)	CCERC	AL	VALID	PSMODE 1	PSMODE 0	OPMODE 2	OPMODE 1	OPMODE 0
000F05E1H		0	0	0	0	0	0	RSTAT	TSTAT
000F05E2H	CLEC(W)	0	0	0	0	0	0	0	0
000F05E2H	CLEC(R)	0	0	0	0	0	LEC2	LEC1	LEC0
000F05E3H	CINFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0
000F05E4H	CERC	TEC [7:0]							
000F05E5H		REPS	REC [7:0]						
000F05E6H	CIE(W)	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0
000F05E7H		0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
000F05E6H	CIE(R)	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
000F05E7H		0	0	0	0	0	0	0	0
000F05E8H	CINTS(W)	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0
000F05E9H		0	0	0	0	0	0	0	0
000F05E8H	CINTS(R)	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0
000F05E9H		0	0	0	0	0	0	0	0

Caution The actual register address is calculated as follows:
Register Address = Global Register Area Offset (CH dependent) + Offset Address as listed in table above

Remark (R) When read
(W) When write

Table 13-18. Bit Configuration of CAN Module Registers (2/2)

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
000F05EAH	CBRP	TQPRS [7:0]							
000F05ECH	CBTR	0	0	0	0	TSEG1 [3:0]			
000F05EDH		0	0	SJW [1:0]		0	TSEG2 [2:0]		
000F05EEH	CLIPT	LIPT [7:0]							
000F05F0H	CRGPT(W)	0	0	0	0	0	0	0	Clear ROVF
000F05F1H		0	0	0	0	0	0	0	0
000F05F0H	CRGPT(R)	0	0	0	0	0	0	RHPM	ROVF
000F05F1H		RGPT [7:0]							
000F05F2H	CLOPT	LOPT [7:0]							
000F05F4H	CTGPT(W)	0	0	0	0	0	0	0	Clear TOVF
000F05F5H		0	0	0	0	0	0	0	0
000F05F4H	CTGPT(R)	0	0	0	0	0	0	THPM	TOVF
000F05F5H		TGPT [7:0]							
000F05F6H	CTS(W)	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN
000F05F7H		0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
000F05F6H	CTS(R)	0	0	0	0	0	TSLOCK	TSSEL	TSEN
000F05F7H		0	0	0	0	0	0	0	0

Caution The actual register address is calculated as follows:
Register Address = Global Register Area Offset (CH dependent) + Offset Address as listed in table above

Remark (R) When read
(W) When write

Table 13-19. Bit Configuration of Message Buffer Registers

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
000F06x0H	CMDB01m	Message data (byte 0)							
000F06x1H		Message data (byte 1)							
000F06x0H	CMDB0m	Message data (byte 0)							
000F06x1H	CMDB1m	Message data (byte 1)							
000F06x2H	CMDB23m	Message data (byte 2)							
000F06x3H		Message data (byte 3)							
000F06x2H	CMDB2m	Message data (byte 2)							
000F06x3H	CMDB3m	Message data (byte 3)							
000F06x4H	CMDB45m	Message data (byte 4)							
000F06x5H		Message data (byte 5)							
000F06x4H	CMDB4m	Message data (byte 4)							
000F06x5H	CMDB5m	Message data (byte 5)							
000F06x6H	CMDB67m	Message data (byte 6)							
000F06x7H		Message data (byte 7)							
000F06x6H	CMDB6m	Message data (byte 6)							
000F06x7H	CMDB7m	Message data (byte 7)							
000F06x8H	CMDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0
000F06x9H	CMCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0
000F06xAH	CMIDLm	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
000F06xBH		ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
000F06xCH	CMIDHm	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
000F06xDH		IDE	0	0	ID28	ID27	ID26	ID25	ID24
000F06xEH	CMCTRLm (W)	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY
000F06xFH		0	0	0	0	Set IE	0	Set TRQ	Set RDY
000F06xEH	CMCTRLm (R)	0	0	0	MOW	IE	DN	TRQ	RDY
000F06xFH		0	0	MUC	0	0	0	0	0

Caution The actual register address is calculated as follows:
Register Address = Global Register Area Offset (CH dependent) + Offset Address as listed in table above

Remarks 1. (R) When read
(W) When write

2. m = 0 to 15

13.6 Bit Set/Clear Function

The CAN control registers include registers whose bits can be set or cleared via the CPU and via the CAN interface. An operation error occurs if the following registers are written directly. Do not write any values directly via bit manipulation, read/modify/write, or direct writing of target values.

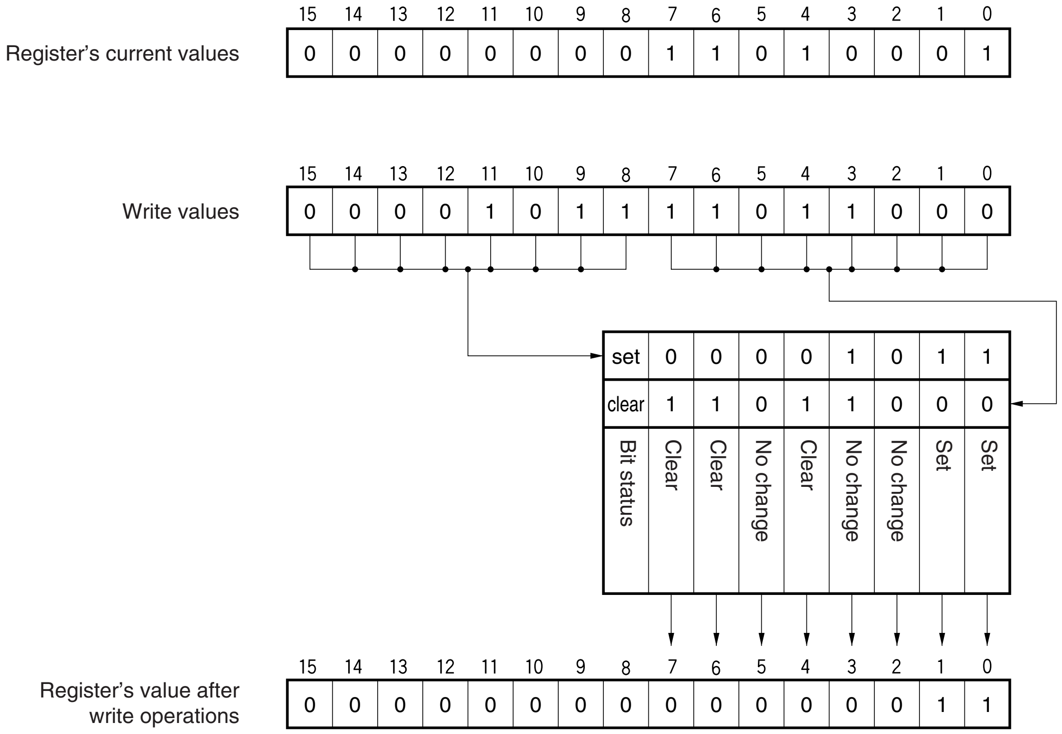
- CAN global control register (CGMCTRL)
- CAN global automatic block transmission control register (CGMABT)
- CAN module control register (CCTRL)
- CAN module interrupt enable register (CIE)
- CAN module interrupt status register (CINTS)
- CAN module receive history list register (CRGPT)
- CAN module transmit history list register (CTGPT)
- CAN module time stamp register (CTS)
- CAN message control register (CMCTRLm)

Remark m = 0 to 15

All the 16 bits in the above registers can be read via the usual method. Use the procedure described in Figure 13-23 below to set or clear the lower 8 bits in these registers.

Setting or clearing of lower 8 bits in the above registers is performed in combination with the higher 8 bits (see the 16-bit data after a write operation in **Figure 13-24**). **Figure 13-23** shows how the values of set bits or clear bits relate to set/clear/no change operations in the corresponding register.

Figure 13-23. Example of Bit Setting/Clearing Operations



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register's current values	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1
Write values	0	0	0	0	1	0	1	1	1	1	0	1	1	0	0	0

set	0	0	0	0	1	0	1	1
clear	1	1	0	1	1	0	0	0
Bit status	Clear	Clear	No change	Clear	No change	No change	Set	Set

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register's value after write operations	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Figure 13-24. 16-Bit Data During Write Operation

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
set 7	set 6	set 5	set 4	set 3	set 2	set 1	set 0	clear 7	clear 6	clear 5	clear 4	clear 3	clear 2	clear 1	clear 0

set n	clear n	Status of bit n after bit set/clear operation
0	0	No change
0	1	0
1	0	1
1	1	No change

Remark n = 0 to 7

13.7 Control Registers

Remark m = 0 to 15

(1) Peripheral clock select register (PCKSEL)

This register is used to select for and supply to each peripheral hardware device the operating clock.
PCKSEL can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Caution Set the PCKSEL register before starting to operate each peripheral hardware device.

Figure 13-25. Format of Peripheral Clock Select Register (PCKSEL)

Address: F00F2H After reset: 00H R/W

Symbol	7	6	5	<4>	3	<2>	1	0
PCKSEL	0	0	0	CANMCKE	0	WUTMCKE	WUTMCK1	WUTMCK0

CANMCKE	Control of CAN controller operating clock
0	Stops supplying operating clock. • CAN controller is a reset state.
1	Supplies operating clock.

<R>

(2) CAN global module control register (CGMCTRL)

The CGMCTRL register is used to control the operation of the CAN module.

Figure 13-26. Format of CAN Global Module Control Register (CGMCTRL) (1/2)

Address: F05C0H After reset: 0000H R/W

(a) Read

	15	14	13	12	11	10	9	8
CGMCTRL	MBON	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	EFSD	GOM

(b) Write

	15	14	13	12	11	10	9	8
CGMCTRL	0	0	0	0	0	0	Set EFSD	Set GOM
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear GOM

(a) Read

MBON	Bit enabling access to message buffer register, transmit/receive history list registers
0	Write access and read access to the message buffer register and the transmit/receive history list registers is disabled.
1	Write access and read access to the message buffer register and the transmit/receive history list registers is enabled.

Cautions 1. While the MBON bit is cleared (to 0), software access to the message buffers (CMDB0m, CMDB1m, CMDB01m, CMDB2m, CMDB3m, CMDB23m, CMDB4m, CMDB5m, CMDB45m, CMDB6m, CMDB7m, CMDB67m, CMDLCm, CMCONFm, CMIDLm, CMIDHm, and CMCTRLm), or registers related to transmit history or receive history (CLOPT, CTGPT, CLIPT, and CRGPT) is disabled.

2. This bit is read-only. Even if 1 is written to MBON while it is 0, the value of MBON does not change, and access to the message buffer registers, or registers related to transmit history or receive history remains disabled.

Remark MBON bit is cleared (to 0) when the CAN module enters CAN sleep mode/CAN stop mode or GOM bit is cleared (to 0).

MBON bit is set (to 1) when the CAN sleep mode/the CAN stop mode is released or GOM bit is set (to 1).

Figure 13-26. Format of CAN Global Module Control Register (CGMCTRL) (2/2)

EFSD	Bit enabling forced shut down
0	Forced shut down by GOM = 0 disabled.
1	Forced shut down by GOM = 0 enabled.

<R> **Caution To request forced shutdown, the GOM bit must be cleared to 0 in a subsequent, immediately following write access after the EFSD bit has been set to 1.**
When other register accesses (reading of a CGMCTRL register is included) by software (interruption) or DMA are executed without clearing the GOM bit immediately after the EFSD bit has been set to 1, the EFSD bit is forcibly cleared to 0, and the forced shutdown request is invalid.

GOM	Global operation mode bit
0	CAN module is disabled from operating.
1	CAN module is enabled to operate.

Caution The GOM bit can be cleared only in the initialization mode or immediately after EFSD bit is set (to 1).

(b) Write

Set EFSD	EFSD bit setting
0	No change in EFSD bit.
1	EFSD bit set to 1.

Set GOM	Clear GOM	GOM bit setting
0	1	GOM bit cleared to 0.
1	0	GOM bit set to 1.
Other than above		No change in GOM bit.

Caution Set GOM bit and EFSD bit always separately.

(3) CAN global module clock select register (CGMCS)

The CGMCS register is used to select the CAN module system clock.

Figure 13-27. Format of CAN Global Module Clock Select Register (CGMCS)

Address: F05CEH After reset: 0FH R/W

	7	6	5	4	3	2	1	0
CGMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0

CCP3	CCP2	CCP1	CCP1	CAN module system clock (f_{CANMOD})
0	0	0	0	f_{CAN}/1
0	0	0	1	f_{CAN}/2
0	0	1	0	f_{CAN}/3
0	0	1	1	f_{CAN}/4
0	1	0	0	f_{CAN}/5
0	1	0	1	f_{CAN}/6
0	1	1	0	f_{CAN}/7
0	1	1	1	f_{CAN}/8
1	0	0	0	f_{CAN}/9
1	0	0	1	f_{CAN}/10
1	0	1	0	f_{CAN}/11
1	0	1	1	f_{CAN}/12
1	1	0	0	f_{CAN}/13
1	1	0	1	f_{CAN}/14
1	1	1	0	f_{CAN}/15
1	1	1	1	f_{CAN}/16 (default value)

Remark f_{CAN}: Clock supplied to CAN (f_{MAIN})

(4) CAN global automatic block transmission control register (CGMABT)

The CGMABT register is used to control the automatic block transmission (ABT) operation.

Figure 13-28. Format of CAN Global Automatic Block Transmission Control Register (CGMABT) (1/2)

Address: F05C6H After reset: 0000H R/W

(a) Read

	15	14	13	12	11	10	9	8
CGMABT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	ABTCLR	ABTTRG

(b) Write

	15	14	13	12	11	10	9	8
CGMABT	0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear ABTTRG

Caution Before changing the normal operation mode with ABT to the initialization mode, be sure to set the CGMABT register to the default value (0000H) and confirm the CGMABT register is surely initialized to the default value(0000H).

(a) Read

ABTCLR	Automatic block transmission engine clear status bit
0	Clearing the automatic transmission engine is completed.
1	The automatic transmission engine is being cleared.

Remarks 1. Set the ABTCLR bit to 1 while the ABTTRG bit is cleared (0). The operation is not guaranteed if the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1.

2. When the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared to 0 as soon as the requested clearing processing is complete.

Figure 13-28. Format of CAN Global Automatic Block Transmission Control Register (CGMABT) (2/2)

ABTTRG	Automatic block transmission status bit
0	Automatic block transmission is stopped.
1	Automatic block transmission is under execution.

Caution Do not set the ABTTRG bit (ABTTRG = 1) in the initialization mode. If the ABTTRG bit is set in the initialization mode, the operation is not guaranteed after the CAN module has entered the normal operation mode with ABT.

Do not set the ABTTRG bit (1) while the CCTRL.TSTAT bit is set (1). Confirm TSTAT = 0 directly in advance before setting ABTTRG bit.

(b) Write

Set ABTCLR	Automatic block transmission engine clear request bit
0	The automatic block transmission engine is in idle state or under operation.
1	Request to clear the automatic block transmission engine. After the automatic block transmission engine has been cleared, automatic block transmission is started from message buffer 0 by setting the ABTTRG bit to 1.

Set ABTTRG	Clear ABTTRG	Automatic block transmission start bit
0	1	Request to stop automatic block transmission.
1	0	Request to start automatic block transmission.
Other than above		No change in ABTTRG bit.

Caution While receiving a message from another node or transmitting the messages other than the ABT messages (message buffer 8 to 15), there is a possibility not to begin immediately the transmission even if the ABTTRG bit is set to 1.

Transmission is not aborted even if the ABTTRG bit is cleared to 0, until the transmission of the ABT message, which is currently being transmitted is completed (successfully or not). After that, the transmission is aborted.

(5) CAN global automatic block transmission delay setting register (CGMABTD)

The CGMABTD register is used to set the interval at which the data of the message buffer assigned to ABT is to be transmitted in the normal operation mode with ABT.

Figure 13-29. Format of CAN Global Automatic Block Transmission Delay Setting Register (CGMABTD)

Address: F05C8H After reset: 00H R/W

	7	6	5	4	3	2	1	0
CGMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

ABTD3	ABTD2	ABTD1	ABTD0	Data frame interval during automatic block transmission (unit: Data bit time (DBT))
0	0	0	0	0 DBT (default value)
0	0	0	1	2^{5} DBT
0	0	1	0	2^{6} DBT
0	0	1	1	2^{7} DBT
0	1	0	0	2^{8} DBT
0	1	0	1	2^{9} DBT
0	1	1	0	2^{10} DBT
0	1	1	1	2^{11} DBT
1	0	0	0	2^{12} DBT
Other than above				Setting prohibited

Cautions 1. Do not change the contents of the CGMABTD register while the ABTTRG bit is set to 1.

2. The timing at which the ABT message is actually transmitted onto the CAN bus differs depending on the status of transmission from the other station or how a request to transmit a message other than an ABT message (message buffers 8 to 15) is made.

(6) CAN module mask register (CMASKaL, CMASKaH) (a = 1, 2, 3, or 4)

The CMASKaL and CMASKaH registers are used to extend the number of receivable messages into the same message buffer by masking part of the ID comparison of a message and invalidating the ID of the masked part.

Figure 13-30. Format of CAN Module Mask Register (CMASKaL, CMASKaH) (a = 1, 2, 3, or 4) (1/2)

- CAN Module Mask 1 Register (CMASK1L, CMASK1H)

Address: F05D0H (CMASK1L), F05D2H (CMASK1H) After reset: Undefined R/W

	15	14	13	12	11	10	9	8
CMASK1L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
CMASK1H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN Module Mask 2 Register (CMASK2L, CMASK2H)

Address: F05D4H (CMASK2L), F05D6H (CMASK2H) After reset: Undefined R/W

	15	14	13	12	11	10	9	8
CMASK2L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
CMASK2H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

Figure 13-30. Format of CAN Module Mask Register (CMASKaL, CMASKaH) (a = 1, 2, 3, or 4) (2/2)

- CAN Module Mask 3 Register (CMASK3L, CMASK3H)

Address: F05D8H (CMASK3L), F05DAH (CMASK3H) After reset: Undefined R/W

	15	14	13	12	11	10	9	8
CMASK3L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
CMASK3H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN Module Mask 4 Register (CMASK4L, CMASK4H)

Address: F05DCH (CMASK4L), F05DEH (CMASK4H) After reset: Undefined R/W

	15	14	13	12	11	10	9	8
CMASK4L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
CMASK4H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

CMID28 to CMID0	Sets Mask Pattern of ID Bit.
0	The ID bits of the message buffer set by the CMID28 to CMID0 bits are compared with the ID bits of the received message frame.
1	The ID bits of the message buffer set by the CMID28 to CMID0 bits are not compared with the ID bits of the received message frame (they are masked).

Remark Masking is always defined by an ID length of 29 bits. If a mask is assigned to a message with a standard ID, CMID17 to CMID0 are ignored. Therefore, only CMID28 to CMID18 of the received ID are masked. The same mask can be used for both the standard and extended IDs.

(7) CAN module control register (CCTRL)

The CCTRL register is used to control the operation mode of the CAN module.

Figure 13-31. Format of CAN Module Control Register (CCTRL) (1/4)

Address: F05E0H After reset: 0000H R/W

(a) Read

	15	14	13	12	11	10	9	8
CCTRL	0	0	0	0	0	0	RSTAT	TSTAT
	7	6	5	4	3	2	1	0
	CCERC	AL	VALID	PSMODE1	PSMODE0	OPMODE2	OPMODE1	OPMODE0

(b) Write

	15	14	13	12	11	10	9	8
CCTRL	Set CCERC	Set AL	0	Set PSMODE1	Set PSMODE0	Set OPMODE2	Set OPMODE1	Set OPMODE0
	7	6	5	4	3	2	1	0
	Clear CCERC	Clear AL	Clear VALID	Clear PSMODE1	Clear PSMODE0	Clear OPMODE2	Clear OPMODE1	Clear OPMODE0

(a) Read

RSTAT	Reception status bit
0	Reception is stopped.
1	Reception is in progress.

Remark - The RSTAT bit is set to 1 under the following conditions (timing).

- The SOF bit of a receive frame is detected
- On occurrence of arbitration loss during a transmit frame

- The RSTAT bit is cleared to 0 under the following conditions (timing)

- When a recessive level is detected at the second bit of the interframe space
- On transition to the initialization mode at the first bit of the interframe space

Figure 13-31. Format of CAN Module Control Register (CCTRL) (2/4)

TSTAT	Transmission status bit
0	Transmission is stopped.
1	Transmission is in progress.

Remark - The TSTAT bit is set to 1 under the following conditions (timing).
- The SOF bit of a transmit frame is detected

- The TSTAT bit is cleared to 0 under the following conditions (timing).
- During transition to bus-off state
- On occurrence of arbitration loss in transmit frame
- On detection of recessive level at the second bit of the interframe space
- On transition to the initialization mode at the first bit of the interframe space

CCERC	Error counter clear bit
0	The CERC and CINFO registers are not cleared in the initialization mode.
1	The CERC and CINFO registers are cleared in the initialization mode.

Remarks 1. The CCERC bit is used to clear the CERC and CINFO registers for re-initialization or forced recovery from the bus-off state. This bit can be set to 1 only in the initialization mode.

2. When the CERC and CINFO registers have been cleared, the CCERC bit is also cleared to 0 automatically.

3. The CCERC bit can be set to 1 at the same time as a request to change the initialization mode to an operation mode is made.

4. The receive data may be corrupted in case of setting the CCERC bit to (1) immediately after entering the INIT mode from self-test mode.

AL	Bit to set operation in case of arbitration loss
0	Re-transmission is not executed in case of an arbitration loss in the single-shot mode.
1	Re-transmission is executed in case of an arbitration loss in the single-shot mode.

Remark The AL bit is valid only in the single-shot mode.

VALID	Valid receive message frame detection bit
0	A valid message frame has not been received since the VALID bit was last cleared to 0.
1	A valid message frame has been received since the VALID bit was last cleared to 0.

Remarks 1. Detection of a valid receive message frame is not dependent upon storage in the receive message buffer (data frame) or transmit message buffer (remote frame).

2. Clear the VALID bit (0) before changing the initialization mode to an operation mode.

3. If only two CAN nodes are connected to the CAN bus with one transmitting a message frame in the normal operation mode and the other in the receive-only mode, the VALID bit is not set to 1 before the transmitting node enters the error passive state, because in receive-only mode no acknowledge is generated.

4. In order to clear the VALID bit, set the Clear VALID bit to 1 first and confirm that the VALID bit is cleared. If it is not cleared, perform clearing processing again.

Figure 13-31. Format of CAN Module Control Register (CCTRL) (3/4)

PSMODE1	PSMODE0	Power save mode
0	0	No power save mode is selected.
0	1	CAN sleep mode
1	0	Setting prohibited
1	1	CAN stop mode

Cautions1. Transition to and from the CAN stop mode must be made via CAN sleep mode. A request for direct transition to and from the CAN stop mode is ignored.

2. The MBON flag of CGMCTRL must be checked after releasing a power save mode, prior to access the message buffers again.

3. CAN Sleep mode requests are kept pending, until cancelled by software or entered on appropriate bus condition (bus idle). Software can check the actual status by reading PSMODE.

<R> **4. The setting change in the Power Save Mode must not combine with operational mode change. These setup must separate and perform a step.**

OPMODE2	OPMODE1	OPMODE0	Operation mode
0	0	0	No operation mode is selected (CAN module is in the initialization mode).
0	0	1	Normal operation mode
0	1	0	Normal operation mode with automatic block transmission function (normal operation mode with ABT)
0	1	1	Receive-only mode
1	0	0	Single-shot mode
1	0	1	Self-test mode
Other than above			Setting prohibited

Cautions 1. Transit to initialization mode or power saving modes may take some time. Be sure to verify the success of mode change by reading the values, before proceeding.

<R> **2. When initialization mode is set up during reception by the Operation Mode, reception of the last which sets DN flag of a message buffer may occur. However, a receiving history list is cleared by the changes which return from initialization mode to the Operation Mode.**

Therefore, before resuming the Operation Mode, it is necessary to clear all the set DN flags about all the effective incoming message buffers.

Remark The OPMODE[2:0] bits are read-only in the CAN sleep mode or CAN stop mode.

(b)Write

Set CCERC	Setting of CCERC bit
1	CCERC bit is set to 1.
Other than above	CCERC bit is not changed.

Set AL	Clear AL	Setting of AL bit
0	1	AL bit is cleared to 0.
1	0	AL bit is set to 1.
Other than above		AL bit is not changed.

Figure 13-31. Format of CAN Module Control Register (CCTRL) (4/4)

Clear VALID	Setting of VALID bit
0	VALID bit is not changed.
1	VALID bit is cleared to 0.

Set PSMODE0	Clear PSMODE0	Setting of PSMODE0 bit
0	1	PSMODE0 bit is cleared to 0.
1	0	PSMODE bit is set to 1.
Other than above		PSMODE0 bit is not changed.

Set PSMODE1	Clear PSMODE1	Setting of PSMODE1 bit
0	1	PSMODE1 bit is cleared to 0.
1	0	PSMODE1 bit is set to 1.
Other than above		PSMODE1 bit is not changed.

Set OPMODE0	Clear OPMODE0	Setting of OPMODE0 bit
0	1	OPMODE0 bit is cleared to 0.
1	0	OPMODE0 bit is set to 1.
Other than above		OPMODE0 bit is not changed.

Set OPMODE1	Clear OPMODE1	Setting of OPMODE1 bit
0	1	OPMODE1 bit is cleared to 0.
1	0	OPMODE1 bit is set to 1.
Other than above		OPMODE1 bit is not changed.

Set OPMODE2	Clear OPMODE2	Setting of OPMODE2 bit
0	1	OPMODE2 bit is cleared to 0.
1	0	OPMODE2 bit is set to 1.
Other than above		OPMODE2 bit is not changed.

(8) CAN module last error code register (CLEC)

The CLEC register provides the error information of the CAN protocol.

Figure 13-32. Format of CAN Module Last Error Code Register (CLEC)

Address: F05E2H After reset: 00H R/W

	7	6	5	4	3	2	1	0
CLEC	0	0	0	0	0	LEC2	LEC1	LEC0

Remarks 1. The contents of the CLEC register are not cleared when the CAN module changes from an operation mode to the initialization mode.

2. If an attempt is made to write a value other than 00H to the CLEC register by software, the access is ignored.

LEC2	LEC1	LEC0	Last CAN protocol error information
0	0	0	No error
0	0	1	Stuff error
0	1	0	Form error
0	1	1	ACK error
1	0	0	Bit error (The CAN module tried to transmit a recessive-level bit as part of a transmit message (except the arbitration field), but the value on the CAN bus is a dominant-level bit.)
1	0	1	Bit error (The CAN module tried to transmit a dominant-level bit as part of a transmit message, ACK bit, error frame, or overload frame, but the value on the CAN bus is a recessive-level bit.)
1	1	0	CRC error
1	1	1	Undefined

(9) CAN module information register (CINFO)

The CINFO register indicates the status of the CAN module.

Figure 13-33. Format of CAN Module Information Register (CINFO)

Address: F05E3H After reset: 00H R

	7	6	5	4	3	2	1	0
CINFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0

BOFF	Bus-off state bit
0	Not bus-off state (transmit error counter ≤ 255) (The value of the transmit counter is less than 256.)
1	Bus-off state (transmit error counter > 255) (The value of the transmit counter is 256 or more.)

TECS1	TECS0	Transmission error counter status bit
0	0	The value of the transmission error counter is less than that of the warning level (<96).
0	1	The value of the transmission error counter is in the range of the warning level (96 to 127).
1	0	Undefined
1	1	The value of the transmission error counter is in the range of the error passive or bus-off state (≥ 128).

RECS1	RECS0	Reception error counter status bit
0	0	The value of the reception error counter is less than that of the warning level (<96).
0	1	The value of the reception error counter is in the range of the warning level (96 to 127).
1	0	Undefined
1	1	The value of the reception error counter is in the error passive range (≥ 128).

(10) CAN module error counter register (CERC)

The CERC register indicates the count value of the transmission/reception error counter.

Figure 13-34. Format of CAN Module Error Counter Register (CERC)

Address: F05E4H After reset: 0000H R

	15	14	13	12	11	10	9	8
CERC	REPS	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

REPS	Reception error passive status bit
0	Reception error counter is not error passive (<128)
1	Reception error counter is error passive range (≥128)

REC6 to REC0	Reception error counter bit
0 to 127	Number of reception errors. These bits reflect the status of the reception error counter. The number of errors is defined by the CAN protocol.

Remark REC[6:0] of the reception error counter are invalid in the reception error passive state (RECS[1:0] = 11B).

TEC7 to TEC0	Transmission error counter bit
0 to 255	Number of transmission errors. These bits reflect the status of the transmission error counter. The number of errors is defined by the CAN protocol.

Remark TEC[7:0] of the transmission error counter are invalid in the bus-off state (BOFF = 1).

(11) CAN module interrupt enable register (CIE)

The CIE register is used to enable or disable the interrupts of the CAN module.

Figure 13-35. Format of CAN Module Interrupt Enable Register (CIE) (1/2)

Address: F05E6H After reset: 0000H R/W

(a) Read

	15	14	13	12	11	10	9	8
CIE	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0

(b) Write

	15	14	13	12	11	10	9	8
CIE	0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
	7	6	5	4	3	2	1	0
	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0

(a) Read

CIE5 to CIE0	CAN module interrupt enable bit
0	Output of the interrupt corresponding to interrupt status register CINTS[5:0] bits is disabled.
1	Output of the interrupt corresponding to interrupt status register CINTS[5:0] bits is enabled.

(b) Write

Set CIE5	Clear CIE5	Setting of CIE5 bit
0	1	CIE5 bit is cleared to 0.
1	0	CIE5 bit is set to 1.
Other than above		CIE5 bit is not changed.

Set CIE4	Clear CIE4	Setting of CIE4 bit
0	1	CIE4 bit is cleared to 0.
1	0	CIE4 bit is set to 1.
Other than above		CIE4 bit is not changed.

Figure 13-35. Format of CAN Module Interrupt Enable Register (CIE) (2/2)

Set CIE3	Clear CIE3	Setting of CIE bit
0	1	CIE3 bit is cleared to 0.
1	0	CIE3 bit is set to 1.
Other than above		CIE3 bit is not changed.

Set CIE2	Clear CIE2	Setting of CIE2 bit
0	1	CIE2 bit is cleared to 0.
1	0	CIE2 bit is set to 1.
Other than above		CIE2 bit is not changed.

Set CIE1	Clear CIE1	Setting of CIE1 bit
0	1	CIE1 bit is cleared to 0.
1	0	CIE1 bit is set to 1.
Other than above		CIE1 bit is not changed.

Set CIE0	Clear CIE0	Setting of CIE0 bit
0	1	CIE0 bit is cleared to 0.
1	0	CIE0 bit is set to 1.
Other than above		CIE0 bit is not changed.

(12) CAN module interrupt status register (CINTS)

The CINTS register indicates the interrupt status of the CAN module.

Figure 13-36. Format of CAN Module Interrupt Status Register (CINTS)

Address: F05E8H After reset: 0000H R/W

(a) Read

	15	14	13	12	11	10	9	8
CINTS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0

(b) Write

	15	14	13	12	11	10	9	8
CINTS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0

(a) Read

CINTS5 to CINTS0	CAN interrupt status bit
0	No related interrupt source event is pending.
1	A related interrupt source event is pending.

Interrupt status bit	Related interrupt source event
CINTS5	Wakeup interrupt from CAN sleep mode[Note]
CINTS4	Arbitration loss interrupt
CINTS3	CAN protocol error interrupt
CINTS2	CAN error status interrupt
CINTS1	Interrupt on completion of reception of valid message frame to message buffer m
CINTS0	Interrupt on normal completion of transmission of message frame from message buffer m

Note The CINTS5 bit is set only when the CAN module is woken up from the CAN sleep mode by a CAN bus operation. The CINTS5 bit is not set when the CAN sleep mode has been released by software.

(b) Write

Clear CINTS5-CINTS0	Setting of CINTS5 to CINTS0 bits
0	CINTS5 to CINTS0 bits are not changed.
1	CINTS5 to CINTS0 bits are cleared to 0.

Caution Please clear the status bit of this register with software when the confirmation of each status is necessary in the interrupt processing, because these bits are not cleared automatically.

(13) CAN module bit rate prescaler register (CBRP)

The CBRP register is used to select the CAN protocol layer basic clock (f_{TQ}). The communication baud rate is set to the CBTR register.

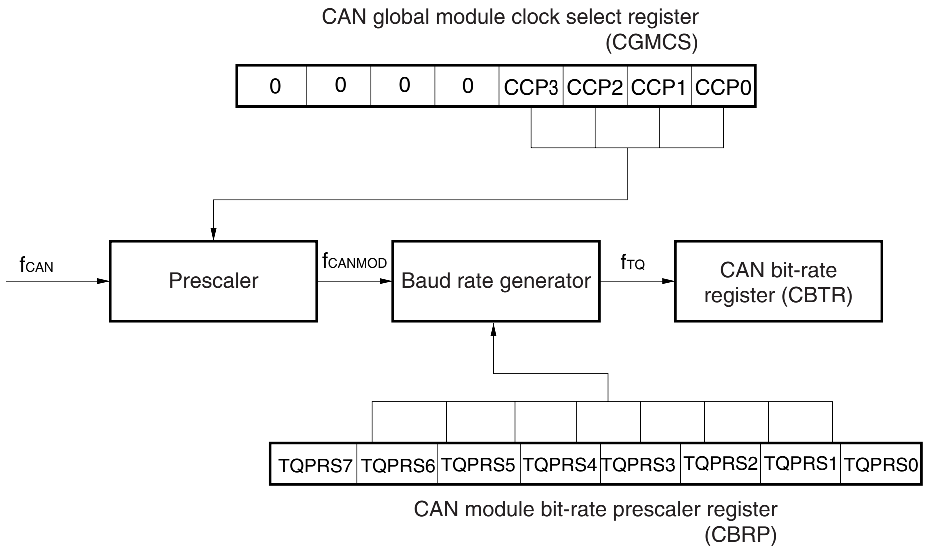
Figure 13-37. Format of CAN Module Bit Rate Prescaler Register (CBRP)

Address: F05EAH After reset: FFH R/W

	7	6	5	4	3	2	1	0
CBRP	TQPRS7	TQPRS6	TQPRS5	TQPRS4	TQPRS3	TQPRS2	TQPRS1	TQPRS0

TQPRS7 to TQPRS0	CAN protocol layer basic system clock (f_{TQ})
0	$f_{CANMOD}/1$
1	$f_{CANMOD}/2$
:	:
n	$f_{CANMOD}/(n+1)$
:	:
255	$f_{CANMOD}/256$ (default value)

Figure 13-38. CAN Global Clock



Caution The CBRP register can be write-accessed only in the initialization mode.

Remark f_{CAN}: Clock supplied to CAN (f_{MAIN})

f_{CANMOD}: CAN module system clock

f_{TQ}: CAN protocol layer basic system clock

(14) CAN module bit rate register (CBTR)

The CBTR register is used to control the data bit time of the communication baud rate.

Figure 13-39. Format of CAN Module Bit Rate Register (CBTR) (1/2)

Address: F05ECH After reset: 370FH R/W

	15	14	13	12	11	10	9	8
CBTR	0	0	SJW1	SJW0	0	TSEG22	TSEG21	TSEG20
	7	6	5	4	3	2	1	0
	0	0	0	0	TSEG13	TSEG12	TSEG11	TSEG10

SJW1	SJW0	Length of synchronization jump width
0	0	1TQ
0	1	2TQ
1	0	3TQ
1	1	4TQ (default value)

TSEG22	TSEG21	TSEG20	Length of time segment 2
0	0	0	1TQ
0	0	1	2TQ
0	1	0	3TQ
0	1	1	4TQ
1	0	0	5TQ
1	0	1	6TQ
1	1	0	7TQ
1	1	1	8TQ (default value)

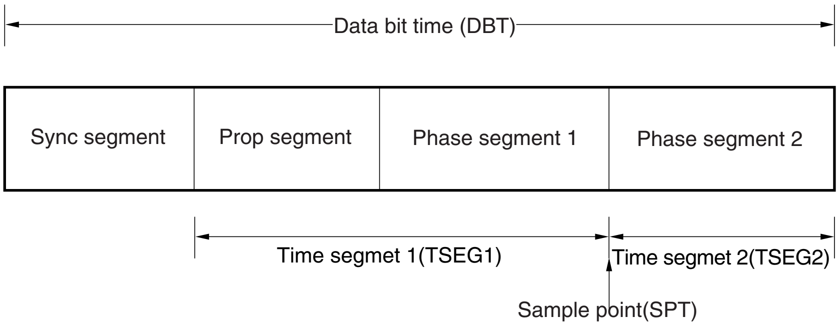
Figure 13-39. Format of CAN Module Bit Rate Register (CBTR) (2/2)

TSEG13	TSEG12	TSEG11	TSEG10	Length of time segment 1
0	0	0	0	Setting prohibited
0	0	0	1	2TQ[Note]
0	0	1	0	3TQ[Note]
0	0	1	1	4TQ
0	1	0	0	5TQ
0	1	0	1	6TQ
0	1	1	0	7TQ
0	1	1	1	8TQ
1	0	0	0	9TQ
1	0	0	1	10TQ
1	0	1	0	11TQ
1	0	1	1	12TQ
1	1	0	0	13TQ
1	1	0	1	14TQ
1	1	1	0	15TQ
1	1	1	1	16TQ (default value)

Note This setting must not be made when the CBRP register = 00H.

Remark TQ = 1/fTQ (fTQ: CAN protocol layer basic system clock)

Figure 13-40. Data Bit Time



(15) CAN module last in-pointer register (CLIPT)

The CLIPT register indicates the number of the message buffer in which a data frame or a remote frame was last stored.

Figure 13-41. Format of CAN Module Last In-pointer Register (CLIPT)

Address: F05EEH After reset: Undefined R

	7	6	5	4	3	2	1	0
CLIPT	LIPT7	LIPT6	LIPT5	LIPT4	LIPT3	LIPT2	LIPT1	LIPT0

LIPT7 to LIPT0	Last in-pointer register (CLIPT)
0 to 15	When the CLIPT register is read, the contents of the element indexed by the last in-pointer (LIPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame was last stored.

Remark The read value of the CLIPT register is undefined if a data frame or a remote frame has never been stored in the message buffer. If the RHPM bit of the CRGPT register is set to 1 after the CAN module has changed from the initialization mode to an operation mode, therefore, the read value of the CLIPT register is undefined.

(16) CAN module receive history list register (CRGPT)

The CRGPT register is used to read the receive history list.

Figure 13-42. Format of CAN Module Receive History List Register (CRGPT) (1/2)

Address: F05F0H After reset: xx02H R/W

(a) Read

	15	14	13	12	11	10	9	8
CRGPT	RGPT7	RGPT6	RGPT5	RGPT4	RGPT3	RGPT2	RGPT1	RGPT0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	RHPM	ROVF

(b) Write

	15	14	13	12	11	10	9	8
CRGPT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear ROVF

(a) Read

RGPT7 to RGPT0	Receive history list get pointer
0 to 15	When the CRGPT register is read, the contents of the element indexed by the receive history list get pointer (RGPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame has been stored.

RHPM[Note]	Receive history list pointer match
0	The receive history list has at least one message buffer number that has not been read.
1	The receive history list has no message buffer numbers that has not been read.

Note The read value of RGPT0 to RGPT7 is invalid when RHPM = 1.

ROVF[Note]	Receive history list overflow bit
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffer in which a new data frame or remote frame has been received and stored are recorded to the receive history list (the receive history list has a vacant element).
1	At least 23 entries have been stored since the host processor has serviced the RHL last time (i.e. read CRGPT). The first 22 entries are sequentially stored while the last entry can have been overwritten whenever newly received message is stored because all buffer numbers are stored at position LIPT-1 when ROVF bit is set. Thus the sequence of receptions can not be recovered completely now.

Note If ROVF is set, RHPM is no longer cleared on message storage, but RHPM is still set, if all entries of CRGPT are read by software.

Figure 13-42. Format of CAN Module Receive History List Register (CRGPT) (2/2)

(b) Write

Clear ROVF	Setting of ROVF bit
0	ROVF bit is not changed.
1	ROVF bit is cleared to 0.

(17) CAN module last out-pointer register (CLOPT)

The CLOPT register indicates the number of the message buffer to which a data frame or a remote frame was transmitted last.

Figure 13-43. Format of CAN Module Last Out-pointer Register (CLOPT)

Address: F05F2H After reset: Undefined R

	7	6	5	4	3	2	1	0
CLOPT	LOPT7	LOPT6	LOPT5	LOPT4	LOPT3	LOPT2	LOPT1	LOPT0

LOPT7 to LOPT0	Last out-pointer of transmit history list (LOPT)
0 to 15	When the CLOPT register is read, the contents of the element indexed by the last out-pointer (LOPT) of the receive history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

Remark The value read from the CLOPT register is undefined if a data frame or remote frame has never been transmitted from a message buffer. If the THPM bit is set to 1 after the CAN module has changed from the initialization mode to an operation mode, therefore, the read value of the CLOPT register is undefined.

(18) CAN module transmit history list register (CTGPT)

The CTGPT register is used to read the transmit history list.

Figure 13-44. Format of CAN Module Transmit History List Register (CTGPT) (1/2)

Address: F05F4H After reset: xx02H R/W

(a) Read

	15	14	13	12	11	10	9	8
CTGPT	TGPT7	TGPT6	TGPT5	TGPT4	TGPT3	TGPT2	TGPT1	TGPT0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	THPM	TOVF

(b) Write

	15	14	13	12	11	10	9	8
CTGPT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear TOVF

(a) Read

TGPT7 to TGPT0	Transmit history list read pointer
0 to 15	When the CTGPT register is read, the contents of the element indexed by the read pointer (TGPT) of the transmit history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

THPM[Note]	Transmit history pointer match
0	The transmit history list has at least one message buffer number that has not been read.
1	The transmit history list has no message buffer number that has not been read.

Note The read value of TGPT0 to TGPT7 is invalid when THPM = 1.

TOVF	Transmit history list overflow bit
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffers to which a new data frame or remote frame has been transmitted are recorded to the transmit history list (the transmit history list has a vacant element).
1	At least 7 entries have been stored since the host processor has serviced the THL last time (i.e. read CTGPT). The first 6 entries are sequentially stored while the last entry can have been overwritten whenever a message is newly transmitted because all buffer numbers are stored at position LOPT-1 when TOVF bit is set. Thus the sequence of transmissions can not be recovered completely now.

Note If TOVF is set, THPM is no longer cleared on message transmission, but THPM is still set, if all entries of CTGPT are read by software.

Remark Transmission from message buffer 0 to 7 is not recorded to the transmit history list in the normal operation mode with ABT.

Figure 13-44. Format of CAN Module Transmit History List Register (CTGPT) (2/2)

(b) Write

Clear TOVF	Setting of TOVF bit
0	TOVF bit is not changed.
1	TOVF bit is cleared to 0.

(19) CAN module time stamp register (CTS)

The CTS register is used to control the time stamp function.

Figure 13-45. Format of CAN Module Time Stamp Register (CTS) (1/2)

Address: F05F6H After reset: 0000H R/W

(a) Read

	15	14	13	12	11	10	9	8
CTS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	TSLOCK	TSSEL	TSEN

(b) Write

	15	14	13	12	11	10	9	8
CTS	0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
	7	6	5	4	3	2	1	0
	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN

Remark The lock function of the time stamp function must not be used when the CAN module is in the normal operation mode with ABT.

(a) Read

TSLOCK	Time stamp lock function enable bit
0	Time stamp lock function stopped. The TSOUT signal is toggled each time the selected time stamp capture event occurs.
1	Time stamp lock function enabled. The TSOUT signal is toggled each time the selected time stamp capture event occurs. However, the TSOUT output signal is locked when a data frame has been correctly received to message buffer 0[Note].

Note The TSEN bit is automatically cleared to 0.

Figure 13-45. Format of CAN Module Time Stamp Register (CTS) (2/2)

TSSEL	Time stamp capture event selection bit
0	The time stamp capture event is SOF.
1	The time stamp capture event is the last bit of EOF.

TSEN	TSOUT signal operation setting bit
0	Disable TSOUT signal toggle operation.
1	Enable TSOUT signal toggle operation.

Remark The signal TSOUT is output from the CAN macro to a timer resource, depending on implementation. See **CHAPTER 6 TIMER ARRAY UNIT**.

(b) Write

Set TSLOCK	Clear TSLOCK	Setting of TSLOCK bit
0	1	TSLOCK bit is cleared to 0.
1	0	TSLOCK bit is set to 1.
Other than above		TSLOCK bit is not changed.

Set TSSEL	Clear TSSEL	Setting of TSSEL bit
0	1	TSSEL bit is cleared to 0.
1	0	TSSEL bit is set to 1.
Other than above		TSSEL bit is not changed.

Set TSEN	Clear TSEN	Setting of TSEN bit
0	1	TSEN bit is cleared to 0.
1	0	TSEN bit is set to 1.
Other than above		TSEN bit is not changed.

(20) CAN message data byte register (CMDBxm) (x = 0 to 7)

The CMDBxm registers are used to store the data of a transmit/receive message. The CMDBzm registers can access the CMDBxm registers in 16-bit units.

Figure 13-46. Format of CAN Message Data Byte Register (CMDBxm) (x = 0 to 7) (1/2)

Address: See **Table 13-16** After reset: Undefined R/W

	15	14	13	12	11	10	9	8
CMDB01m	MDATA0115	MDATA0114	MDATA0113	MDATA0112	MDATA0111	MDATA0110	MDATA019	MDATA018
	7	6	5	4	3	2	1	0
	MDATA017	MDATA016	MDATA015	MDATA014	MDATA013	MDATA012	MDATA011	MDATA010

	7	6	5	4	3	2	1	0
CMDB0m	MDATA07	MDATA06	MDATA05	MDATA04	MDATA03	MDATA02	MDATA01	MDATA00

	7	6	5	4	3	2	1	0
CMDB1m	MDATA17	MDATA16	MDATA15	MDATA14	MDATA13	MDATA12	MDATA11	MDATA10

	15	14	13	12	11	10	9	8
CMDB23m	MDATA2315	MDATA2314	MDATA2313	MDATA2312	MDATA2311	MDATA2310	MDATA239	MDATA238
	7	6	5	4	3	2	1	0
	MDATA237	MDATA236	MDATA235	MDATA234	MDATA233	MDATA232	MDATA231	MDATA230

	7	6	5	4	3	2	1	0
CMDB2m	MDATA27	MDATA26	MDATA25	MDATA24	MDATA23	MDATA22	MDATA21	MDATA20

	7	6	5	4	3	2	1	0
CMDB3m	MDATA37	MDATA36	MDATA35	MDATA34	MDATA33	MDATA32	MDATA31	MDATA30

	15	14	13	12	11	10	9	8
CMDB45m	MDATA4515	MDATA4514	MDATA4513	MDATA4512	MDATA4511	MDATA4510	MDATA459	MDATA458
	7	6	5	4	3	2	1	0
	MDATA457	MDATA456	MDATA455	MDATA454	MDATA453	MDATA452	MDATA451	MDATA450

	7	6	5	4	3	2	1	0
CMDB4m	MDATA47	MDATA46	MDATA45	MDATA44	MDATA43	MDATA42	MDATA41	MDATA40

	7	6	5	4	3	2	1	0
CMDB5m	MDATA57	MDATA56	MDATA55	MDATA54	MDATA53	MDATA52	MDATA51	MDATA50

Figure 13-46. Format of CAN Message Data Byte Register (CMDBxm) (x = 0 to 7) (2/2)

	15	14	13	12	11	10	9	8
CMDB67m	MDATA6715	MDATA6714	MDATA6713	MDATA6712	MDATA6711	MDATA6710	MDATA679	MDATA678
	7	6	5	4	3	2	1	0
	MDATA677	MDATA676	MDATA675	MDATA674	MDATA673	MDATA672	MDATA671	MDATA670

	7	6	5	4	3	2	1	0
CMDB6m	MDATA67	MDATA66	MDATA65	MDATA64	MDATA63	MDATA62	MDATA61	MDATA60

	7	6	5	4	3	2	1	0
CMDB7m	MDATA77	MDATA76	MDATA75	MDATA74	MDATA73	MDATA72	MDATA71	MDATA70

(21) CAN message data length register m (CMDLCm)

The CMDLCm register is used to set the number of bytes of the data field of a message buffer.

Figure 13-47. Format of CAN Message Data Length Register m (CMDLCm)

Address: See **Table 13-16** After reset: 0000xxxxB R/W

	7	6	5	4	3	2	1	0
CMDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0

MDLC3	MDLC2	MDLC1	MDLC0	Data length of transmit/receive message
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
1	0	0	1	Setting prohibited (If these bits are set during transmission, 8-byte data is transmitted regardless of the set DLC value when a data frame is transmitted. However, the DLC actually transmitted to the CAN bus is the DLC value set to this register.)[Note]
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Note The data and DLC value actually transmitted to CAN bus are as follows.

Type of transmit frame	Length of transmit data	DLC transmitted
Data frame	Number of bytes specified by DLC (However, 8 bytes if DLC ≥ 8)	MDLC[3:0]
Remote frame	0 bytes	

Cautions 1. Be sure to set bits 7 to 4 0000B.

2. Receive data is stored in as many CMDBxm as the number of bytes (however, the upper limit is 8) corresponding to DLC of the received frame. CMDBxm in which no data is stored is undefined.

(22) CAN message configuration register (CMCONFm)

The CMCONFm register is used to specify the type of the message buffer and to set a mask.

Figure 13-48. Format of CAN Message Configuration Register (CMCONFm) (1/2)

Address: See **Table 13-16** After reset: Undefined R/W

	7	6	5	4	3	2	1	0
CMCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0

Caution Be sure to write 0 to bits 2 and 1.

OWS	Overwrite control bit
0	The message buffer that has already received a data frame[Note] is not overwritten by a newly received data frame. The newly received data frame is discarded.
1	The message buffer that has already received a data frame[Note] is overwritten by a newly received data frame.

Note The "message buffer that has already received a data frame" is a receive message buffer whose DN bit has been set to 1.

Remark A remote frame is received and stored, regardless of the setting of OWS bit and DN bit. A remote frame that satisfies the other conditions (ID matches, RTR = 0, TRQ = 0) is always received and stored in the corresponding message buffer (interrupt generated, DN flag set, MDLC[3:0] bits updated, and recorded to the receive history list).

RTR	Remote frame request bit[Note]
0	Transmit a data frame.
1	Transmit a remote frame.

Note The RTR bit specifies the type of message frame that is transmitted from a message buffer defined as a transmit message buffer. Even if a valid remote frame has been received, RTR of the transmit message buffer that has received the frame remains cleared to 0. Even if a remote frame whose ID matches has been received from the CAN bus with the RTR bit of the transmit message buffer set to 1 to transmit a remote frame, that remote frame is not received or stored (interrupt generated, DN flag set, MDLC[3:0] bits updated, and recorded to the receive history list).

MT2	MT1	MT0	Message buffer type setting bit
0	0	0	Transmit message buffer
0	0	1	Receive message buffer (no mask setting)
0	1	0	Receive message buffer (mask 1 set)
0	1	1	Receive message buffer (mask 2 set)
1	0	0	Receive message buffer (mask 3 set)
1	0	1	Receive message buffer (mask 4 set)
Other than above			Setting prohibited

Figure 13-48. Format of CAN Message Configuration Register (CMCONFm) (2/2)

MA0	Message buffer assignment bit
0	Message buffer not used.
1	Message buffer used.

(23) CAN message ID register m (CMIDLm, CMIDHm)

The CMIDLm and CMIDHm registers are used to set an identifier (ID).

Figure 13-49. Format of CAN Message ID Register m (CMIDLm, CMIDHm)

Address: See **Table 13-16** After reset: Undefined R/W

	15	14	13	12	11	10	9	8
CMIDLm	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
	7	6	5	4	3	2	1	0
	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

	15	14	13	12	11	10	9	8
CMIDHm	IDE	0	0	ID28	ID27	ID26	ID25	ID24
	7	6	5	4	3	2	1	0
	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

IDE	Format mode specification bit
0	Standard format mode (ID28 to ID18: 11 bits)[Note]
1	Extended format mode (ID28 to ID0: 29 bits)

Note The ID17 to ID0 bits are not used.

ID28 to ID0	Message ID
ID28 to ID18	Standard ID value of 11 bits (when IDE = 0)
ID28 to ID0	Extended ID value of 29 bits (when IDE = 1)

Cautions 1. Be sure to write 0 to bits 14 and 13 of the CMIDHm register.

2. Be sure to align the ID value according to the given bit positions into this registers. Note that for standard ID, the ID value must be shifted to fit into ID28 to ID11 bit positions.

(24) CAN message control register m (CMCTRLm)

The CMCTRLm register is used to control the operation of the message buffer.

Figure 13-50. Format of CAN Message Control Register m (CMCTRLm) (1/3)

Address: **Table 13-16**. After reset: 00x00000 R/W
000xx000B

(a) Read

	15	14	13	12	11	10	9	8
CMCTRLm	0	0	MUC	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	MOW	IE	DN	TRQ	RDY

(b) Write

	15	14	13	12	11	10	9	8
CMCTRLm	0	0	0	0	Set IE	0	Set TRQ	Set RDY
	7	6	5	4	3	2	1	0
	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY

(a) Read

MUC[Note]	Message buffer data updating bit
0	The CAN module is not updating the message buffer (reception and storage).
1	The CAN module is updating the message buffer (reception and storage).

Note The MUC bit is undefined until the first reception and storage is performed.

MOW	Message buffer overwrite status bit
0	The message buffer is not overwritten by a newly received data frame.
1	The message buffer is overwritten by a newly received data frame.

Remark MOW bit is not set to 1 even if a remote frame is received and stored in the transmit message buffer with DN = 1.

IE	Message buffer interrupt request enable bit
0	Receive message buffer: Valid message reception completion interrupt disabled. Transmit message buffer: Normal message transmission completion interrupt disabled.
1	Receive message buffer: Valid message reception completion interrupt enabled. Transmit message buffer: Normal message transmission completion interrupt enabled.

Figure 13-50. Format of CAN Message Control Register m (CMCTRLm) (2/3)

DN	Message buffer data updating bit
0	A data frame or remote frame is not stored in the message buffer.
1	A data frame or remote frame is stored in the message buffer.

TRQ	Message buffer transmission request bit
0	No message frame transmitting request that is pending or being transmitted is in the message buffer.
1	The message buffer is holding transmission of a message frame pending or is transmitting a message frame.

Caution Do not set the TRQ bit and the RDY bit (1) at the same time. Set the RDY bit (1) before setting the TRQ bit.

RDY	Message buffer ready bit
0	The message buffer can be written by software. The CAN module cannot write to the message buffer.
1	Writing the message buffer by software is ignored (except a write access to the RDY, TRQ, DN, and MOW bits). The CAN module can write to the message buffer.

Cautions1. Do not clear the RDY bit (0) during message transmission.
Follow the transmission abort process about clearing the RDY bit (0) for redefinition of the message buffer.

2. Clear again when RDY bit is not cleared even if this bit is cleared.

3. Be sure that RDY is cleared before writing to the message buffer registers. Perform this confirmation by reading back the RDY bit.
However, setting the TRQ bit, clearing the DN bit, setting the RDY bit or clearing the MOW bit of the CMCTRLm register need not be confirmed.

(b) Write

Clear MOW	Setting of MOW bit
0	MOW bit is not changed.
1	MOW bit is cleared to 0.

Set IE	Clear IE	Setting of IE bit
0	1	IE bit is cleared to 0.
1	0	IE bit is set to 1.
Other than above		IE bit is not changed.

Caution Set IE bit and RDY bit always separately.

Clear DN	Setting of DN bit
0	DN bit is not changed.
1	DN bit is cleared to 0.

Cautions 1. Do not set the DN bit to 1 by software. Be sure to write 0 to bit 10.

<R> **2. When DN bit is cleared (0) by the end of the received arbitration field, the message buffer is the search target for storing a receiving frame.**

Figure 13-50. Format of CAN Message Control Register m (CMCTRLm) (3/3)

Set TRQ	Clear TRQ	Setting of TRQ bit
0	1	TRQ bit is cleared to 0.
1	0	TRQ bit is set to 1.
Other than above		TRQ bit is not changed.

Caution While receiving a message from another node or transmitting the messages, there is a possibility of not to begin immediately the transmission even if the TRQ bit is set to 1.

The transmission is not aborted even if the TRQ bit is cleared to 0. The transmission is continued if a message is currently being transmitted and until the transmission is completed (successfully or not).

Set RDY	Clear RDY	Setting of RDY bit
0	1	RDY bit is cleared to 0.
1	0	RDY bit is set to 1.
Other than above		RDY bit is not changed.

Caution Set IE bit and RDY bit always separately.

(25) Serial communication pin select register (STSEL)

The STSEL register is used to switch the input source to the timer array unit and the LIN-UARTn and CAN communication pins.

This register can be read or written in 1-bits unit or 8-bit units.

With the 78K0R/Hx3, the STSEL register can be used to select which set of CTxD, CRxD pins provided at two different ports to use.

Figure 13-51. Format of Serial Communication Pin Select Register (STSEL)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
STSEL	STSLIN1	TMCAN	STSIIC11	STSCSI00	TM30K	TMLIN1	TMLIN0	0

STSLIN1	Serial communication pin selection			
	LIN-UART pin		CAN pin	
	LTxD1	LRxD1/INTPLR1	CTxD	CRxD
0	P10	P11	P72	P73
1	P72	P73	P10	P11

TMCAN	Switching source of input to channel 4 of timer array unit 1
0	TI14 pin input (pin input selected by using the TI14 bit)
1	TSOUT input (CAN time stamp function)

Remark During LIN communication, when in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B), select the input signal of the serial data input pin (LRxDn) as a timer input by setting TMLINn =1.

(26) Port mode registers 1, 7 (PM1, PM7)

The PM1 and PM7 registers are used to set ports 1 and 7 to input or output in 1-bit units.

When using the P10/TI10/$\overline{\text{SCK10}}$/TO00/CTxD/LTxD1, and P72/KR2/CTxD/LTxD1 pins for serial data output, clear the PM1_0, and PM7_2 bits to “0”, and set the output latches of P1_0, and P7_2 to “1”.

When using the P11/TI02/SI10/LRxD1/INTPLR1/CRxD/TO02, and P73/KR3/CRxD/LRxD1/INTPLR1 pins for serial data input, set the PM1_1, and PM7_3 bits to “1”. At this time, the output latches of P1_1, and P7_3 may be “0” or “1”.

The PM1 and PM7 registers can be set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark The pins mounted depend on the product. See **1.4 Ordering Information** and **2.1 Pin Function List**.

<R> **Figure 13-52. Format of Port Mode Registers 1, 7 (PM1, PM7)**

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM1_7	PM1_6	PM1_5	PM1_4	PM1_3	PM1_2	PM1_1	PM1_0

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM7_7	PM7_6	PM7_5	PM7_4	PM7_3	PM7_2	PM7_1	PM7_0

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

13.8 CAN Controller Initialization

13.8.1 Initialization of CAN module

Before the CAN module operation is enabled, the CAN module system clock needs to be determined by setting the CCP[3:0] bits of the CGMCS register by software. Do not change the setting of the CAN module system clock after CAN module operation is enabled.

The CAN module is enabled by setting the GOM bit of the CGMCTRL register.

For the procedure of initializing the CAN module, see **13.16 Operation of CAN Controller**.

13.8.2 Initialization of message buffer

After the CAN module is enabled, the message buffers contain undefined values. A minimum initialization for all the message buffers, even for those not used in the application, is necessary before switching the CAN module from the initialization mode to one of the operation modes.

- Clear the RDY, TRQ, and DN bits of the CMCTRLm register to 0.
- Clear the MA0 bit of the CMCONFm register to 0.

Remark m = 0 to 15

13.8.3 Redefinition of message buffer

Redefining a message buffer means changing the ID and control information of the message buffer while a message is being received or transmitted, without affecting other transmission/reception operations.

(1) To redefine message buffer in initialization mode

Place the CAN module in the initialization mode once and then change the ID and control information of the message buffer in the initialization mode. After changing the ID and control information, set the CAN module in an operation mode.

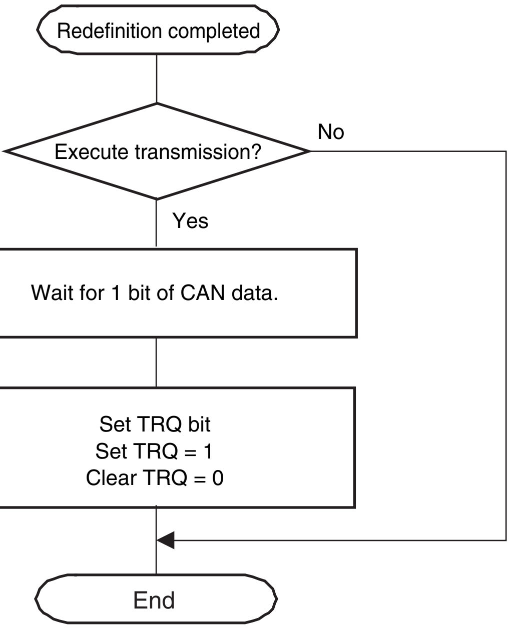
(2) To redefine message buffer during reception

Perform redefinition as shown in Figure 13-66.

(3) To redefine message buffer during transmission

To rewrite the contents of a transmit message buffer to which a transmission request has been set, perform transmission abort processing (see **13.10.4 (1) Transmission abort process except for in normal operation mode with automatic block transmission (ABT)** and **13.10.4 (2) Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)**. Confirm that transmission has been aborted or completed, and then redefine the message buffer. After redefining the transmit message buffer, set a transmission request using the procedure described below. When setting a transmission request to a message buffer that has been redefined without aborting the transmission in progress, however, the 1-bit wait time is not necessary.

Figure 13-53. Setting Transmission Request (TRQ) to Transmit Message Buffer After Redefining



Cautions 1. When a message is received, reception filtering is performed in accordance with the ID and mask set to each receive message buffer. If the procedure in Figure 13-66 is not observed, the contents of the message buffer after it has been redefined may contradict the result of reception (result of reception filtering). If this happens, check that the ID and IDE received first and stored in the message buffer following redefinition are those stored after the message buffer has been redefined. If no ID and IDE are stored after redefinition, redefine the message buffer again.

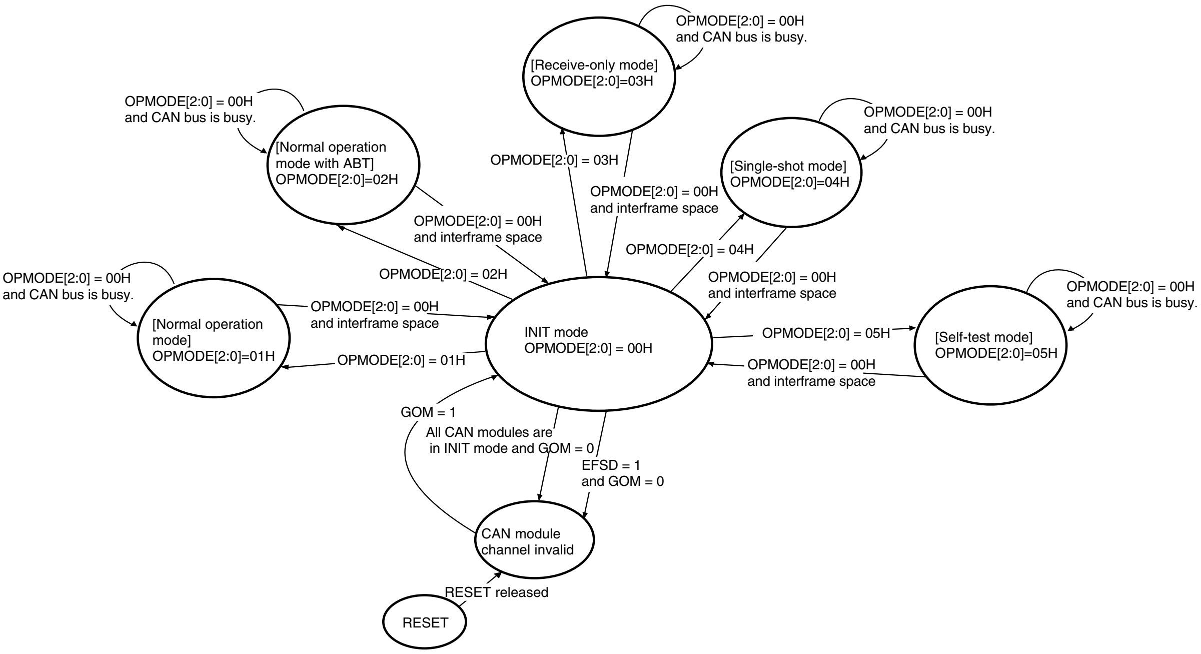
2. When a message is transmitted, the transmission priority is checked in accordance with the ID, IDE, and RTR bits set to each transmit message buffer to which a transmission request was set. The transmit message buffer having the highest priority is selected for transmission. If the procedure in Figure 13-53 is not observed, a message with an ID not having the highest priority may be transmitted after redefinition.

13.8.4 Transition from initialization mode to operation mode

The CAN module can be switched to the following operation modes.

- Normal operation mode
- Normal operation mode with ABT
- Receive-only mode
- Single-shot mode
- Self-test mode

Figure 13-54. Transition to Operation Modes



The transition from the initialization mode to an operation mode is controlled by the bit string OPMODE[2:0] in the CCTRL register.

Changing from one operation mode into another requires shifting to the initialization mode in between. Do not change one operation mode to another directly; otherwise the operation will not be guaranteed.

Requests for transition from the operation mode to the initialization mode are held pending when the CAN bus is not in the interframe space (i.e., frame reception or transmission is in progress), and the CAN module enters the initialization mode at the first bit in the interframe space (the value of OPMODE[2:0] are changed to 00H). After issuing a request to change the mode to the initialization mode, read the OPMODE[2:0] bits until their value becomes 000B to confirm that the module has entered the initialization mode (see **Figure 13-64**).

13.8.5 Resetting error counter CERC of CAN module

If it is necessary to reset the CAN module error counter CERC and the CAN module information register CINFO when re-initialization or forced recovery from the bus-off state is made, set the CCERC bit of the CCTRL register to 1 in the initialization mode. When this bit is set to 1, the CAN module error counter CERC and the CAN module information register CINFO are cleared to their default values.

13.9 Message Reception

13.9.1 Message reception

In all the operation modes, the complete message buffer area is analyzed to find a suitable buffer to store a newly received message. All message buffers satisfying the following conditions are included in that evaluation (RX-search process).

- Used as a message buffer
  (MA0 bit of CMCONFm register set to 1B.)
- Set as a receive message buffer
  (MT[2:0] bits of CMCONFm register set to 001B, 010B, 011B, 100B, or 101B.)
- Ready for reception
  (RDY bit of CMCTRLm register set to 1.)

When two or more message buffers of the CAN module receive a message, the message is stored according to the priority explained below. The message is always stored in the message buffer with the highest priority, not in a message buffer with a low priority. For example, when an unmasked receive message buffer and a receive message buffer linked to mask 1 have the same ID, the received message is not stored in the message buffer linked to mask 1, even if that message buffer has not received a message and a message has already been received in the unmasked receive message buffer. In other words, when a condition has been set to store a message in two or more message buffers with different priorities, the message buffer with the highest priority always stores the message; the message is not stored in message buffers with a lower priority. This also applies when the message buffer with the highest priority is unable to receive and store a message (i.e., when DN = 1 indicating that a message has already been received, but rewriting is disabled because OWS = 0). In this case, the message is not actually received and stored in the candidate message buffer with the highest priority, but neither is it stored in a message buffer with a lower priority.

Priority	Storing Condition if Same ID Is Set	
1 (high)	Unmasked message buffer	DN = 0
		DN = 1 and OWS = 1
2	Message buffer linked to mask 1	DN = 0
		DN = 1 and OWS = 1
3	Message buffer linked to mask 2	DN = 0
		DN = 1 and OWS = 1
4	Message buffer linked to mask 3	DN = 0
		DN = 1 and OWS = 1
5(low)	Message buffer linked to mask 4	DN = 0
		DN = 1 and OWS = 1

Remark m = 0 to 15

13.9.2 Receive data read

To keep data consistency when reading CAN message buffers, perform the data reading according to Figure 13-77 to 13-79.

During message reception, the CAN module sets DN of the CMCTRLm register two times: at the beginning of the storage process of data to the message buffer, and again at the end of this storage process. During this storage process, the MUC bit of the CMCTRLm register of the message buffer is set. (see **Figure 13-55**.)

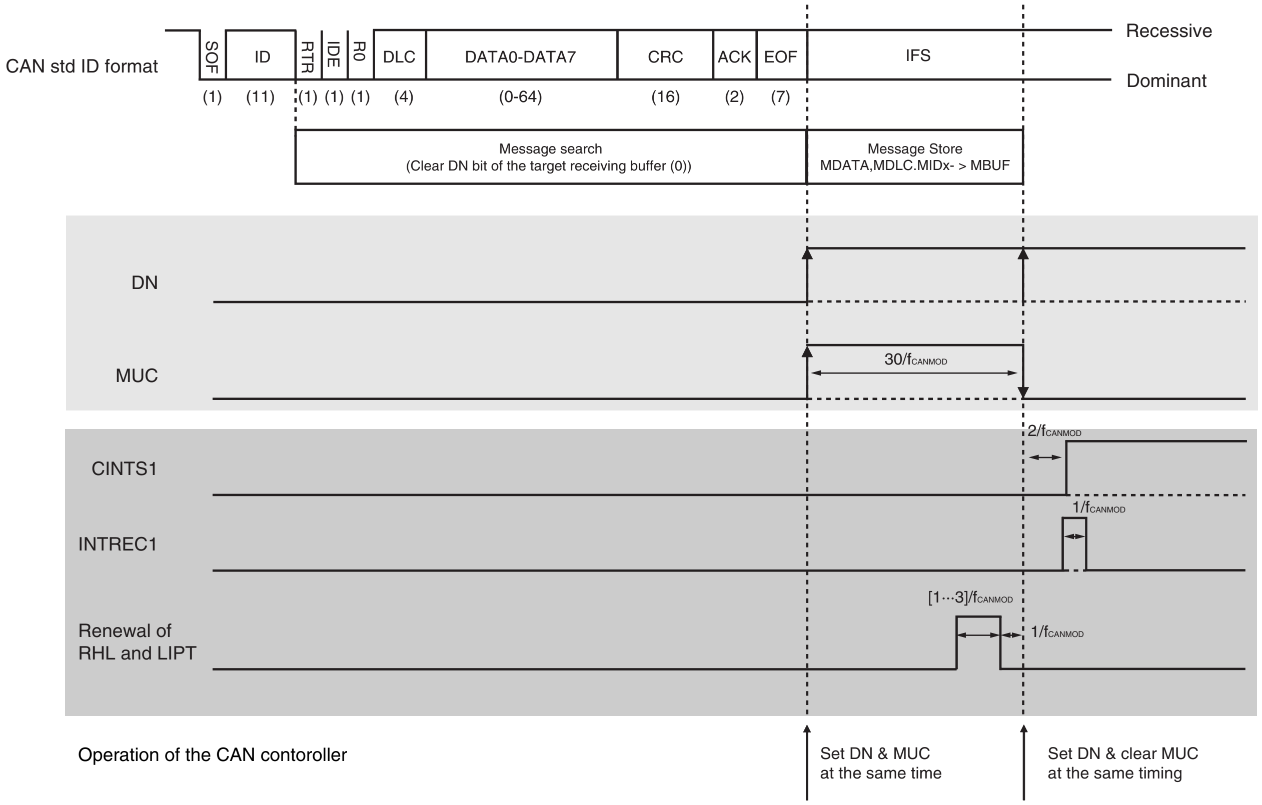
The receive history list is also updated just before the storage process. In addition, during storage process (MUC = 1), the RDY bit of the CMCTRL register of the message buffer is locked to avoid the coincidental data WR by CPU. Note the storage process may be disturbed (delayed) when the CPU accesses the message buffer.

<R> It is necessary to clear DN bit of a buffer to certainly store a message in a message buffer, before message search processing is started (before ID of a frame is outputted on Bus).

This is the 15th bit from after EOF of the last frame at the shortest.

It recommends using more message buffers for frame reception than one piece for the CAN frame to continue on a bus, appear and receive it certainly.

<R> **Figure 13-55. Receiving Timing**



Remark f_{CANMOD}: CAN module system clock

13.9.3 Receive history list function

The receive history list (RHL) function records in the receive history list the number of the receive message buffer in which each data frame or remote frame was received and stored. The RHL consists of storage elements equivalent to up to 23 messages, the last in-message pointer (LIPT) with the corresponding CLIPT register and the receive history list get pointer (RGPT) with the corresponding CRGPT register.

The RHL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The CLIPT register holds the contents of the RHL element indicated by the value of the LIPT pointer minus 1. By reading the CLIPT register, therefore, the number of the message buffer that received and stored a data frame or remote frame first can be checked. The LIPT pointer is utilized as a write pointer that indicates to what part of the RHL a message buffer number is recorded. Any time a data frame or remote frame is received and stored, the corresponding message buffer number is recorded to the RHL element indicated by the LIPT pointer. Each time recording to the RHL has been completed, the LIPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

The RGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the RHL. This pointer indicates the first RHL element that the CPU has not read yet. By reading the CRGPT register by software, the number of a message buffer that has received and stored a data frame or remote frame can be read. Each time a message buffer number is read from the CRGPT register, the RGPT pointer is automatically incremented.

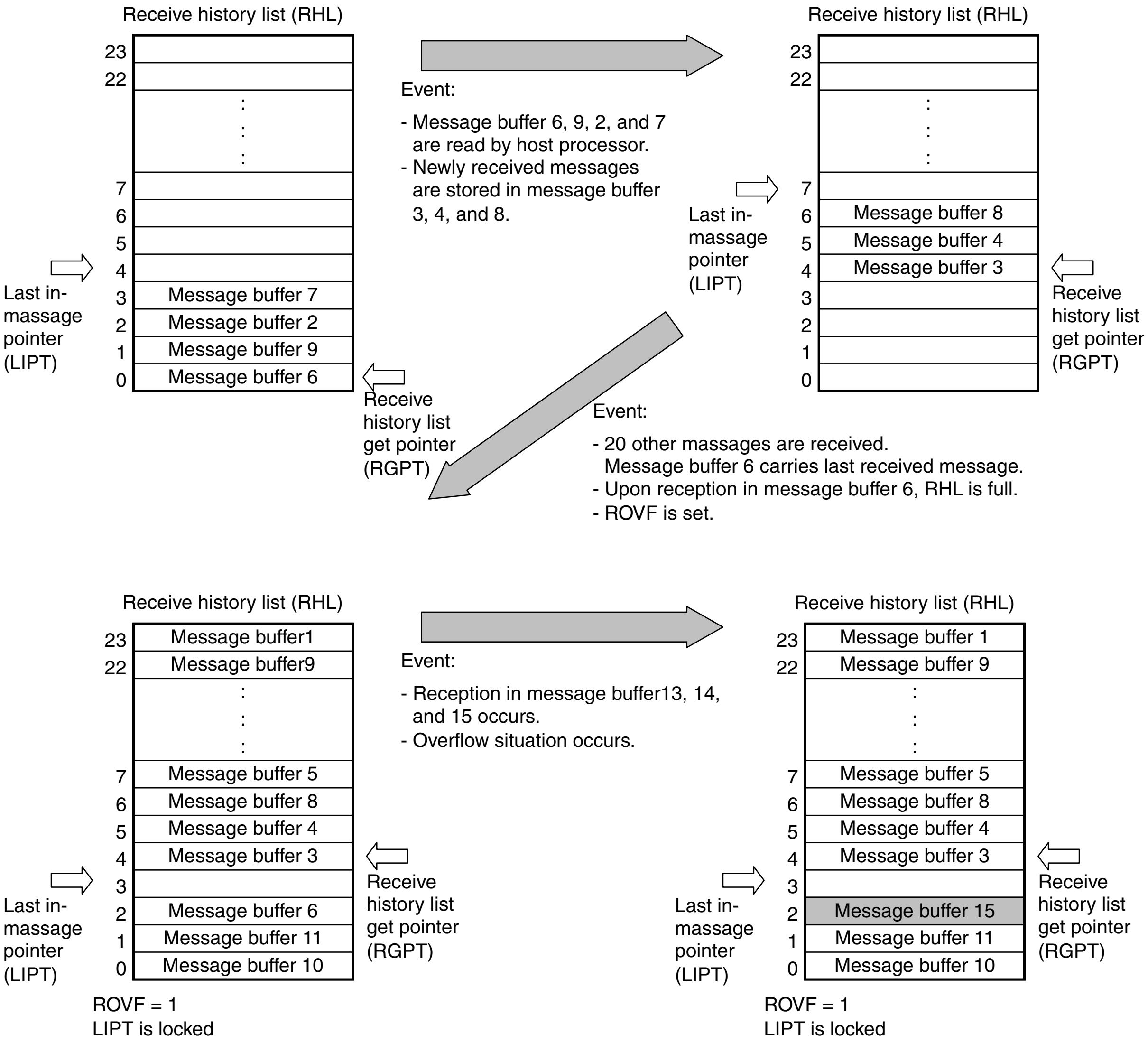
If the value of the RGPT pointer matches the value of the LIPT pointer, the RHPM bit (receive history list pointer match) of the CRGPT register is set to 1. This indicates that no message buffer number that has not been read remains in the RHL. If a new message buffer number is recorded, the LIPT pointer is incremented and because its value no longer matches the value of the RGPT pointer, the RHPM bit is cleared. In other words, the numbers of the unread message buffers exist in the RHL.

If the LIPT pointer is incremented and matches the value of the RGPT pointer minus 1, the ROVF bit (receive history list overflow) of the CRGPT register is set to 1. This indicates that the RHL is full of numbers of message buffers that have not been read. When further message reception and storing occur, the last recorded message buffer number is overwritten by the number of the message buffer that received and stored the new message. In this case, after the ROVF bit has been set (1), the recorded message buffer numbers in the RHL do not completely reflect the chronological order. However messages itself are not lost and can be located by CPU search in message buffer memory with the help of the DN bit.

Caution If the history list is in the overflow condition (ROVF is set), reading the history list contents is still possible, until the history list is empty (indicated by RHPM flag set). Nevertheless, the history list remains in the overflow condition, until ROVF is cleared by software. If ROVF is not cleared, the RHPM flag will also not be updated (cleared) upon a message storage of newly received frame. This may lead to the situation, that RHPM indicates an empty history list, although a reception has taken place, while the history list is in the overflow state (ROVF and RHPM are set).

As long as the RHL contains 23 or less entries the sequence of occurrence is maintained. If more receptions occur without reading the RHL by the host processor, complete sequence of receptions can not be recovered.

Figure 13-56. Receive History List



Event:

- Message buffer 6, 9, 2, and 7 are read by host processor.
- Newly received messages are stored in message buffer 3, 4, and 8.

Event:

- 20 other massages are received. Message buffer 6 carries last received message.
- Upon reception in message buffer 6, RHL is full.
- ROVF is set.

Event:

- Reception in message buffer13, 14, and 15 occurs.
- Overflow situation occurs.

ROVF = 1 denotes that LIPT equals RGPT−1 while message buffer number stored to element indicated by LIPT−1.

13.9.4 Mask function

For any message buffer, which is used for reception, the assignment to one of four global reception masks (or no mask) can be selected.

By using the mask function, the message ID comparison can be reduced by masked bits, herewith allowing the reception of several different IDs into one buffer.

While the mask function is in effect, an identifier bit that is defined to be “1” by a mask in the received message is not compared with the corresponding identifier bit in the message buffer.

However, this comparison is performed for any bit whose value is defined as “0” by the mask.

For example, let us assume that all messages that have a standard-format ID, in which bits ID27 to ID25 are “0” and bits ID24 and ID22 are “1”, are to be stored in message buffer 14. The procedure for this example is shown below.

<1> Identifier to be stored in message buffer

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

x = don’t care

<2> Identifier to be configured in message buffer 14 (example)
(using CAN message ID registers L14 and H14 (CMIDL14 and CMIDH14))

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
x	x	x	x	x	x	x	x	x	x	x

ID6	ID5	ID4	ID3	ID2	ID1	ID0
x	x	x	x	x	x	x

ID with ID27 to ID25 cleared to “0” and ID24 and ID22 set to “1” is registered (initialized) to message buffer 14.

Remark Message buffer 14 is set as a standard format identifier that is linked to mask 1 (MT[2:0] of CMCONF14 register are set to 010B).

<3> Mask setting for CAN module 1 (mask 1) (Example)
(Using CAN module mask 1 registers L and H (CMASK1L and CMASK1H))

CMID28	CMID27	CMID26	CMID25	CMID24	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18
1	0	0	0	0	1	0	1	1	1	1

CMID17	CMID16	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	CMID7
1	1	1	1	1	1	1	1	1	1	1

CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
1	1	1	1	1	1	1

1: Not compared (masked)
0: Compared

The CMID27 to CMID24 and CMID22 bits are cleared to “0”, and CMID28, CMID23, and CMID21 to CMID0 bits are set to “1”.

13.9.5 Multi buffer receive block function

The multi buffer receive block (MBRB) function is used to store a block of data in two or more message buffers sequentially with no CPU interaction, by setting the same ID to two or more message buffers with the same message buffer type.

Suppose, for example, the same message buffer type is set to 5 message buffers, message buffers 10 to 14, and the same ID is set to each message buffer. If the first message whose ID matches the ID of the message buffers is received, it is stored in message buffer 10. At this point, the DN bit of message buffer 10 is set, prohibiting overwriting the message buffer when subsequent messages are received.

If the next message with a matching ID is received, it is received and stored in message buffer 11. Each time a message with a matching ID is received, it is sequentially (in the ascending order) stored in message buffers 12, 13, and 14. Even when a data block consisting of multiple messages is received, the messages can be stored and received without overwriting the previously received matching-ID data.

Whether a data block has been received and stored can be checked by setting the IE bit of the CMCTRLm register of each message buffer. For example, if a data block consists of k messages, k message buffers are initialized for reception of the data block. The IE bit in message buffers 0 to (k-2) is cleared to 0 (interrupts disabled), and the IE bit in message buffer k-1 is set to 1 (interrupts enabled). In this case, a reception completion interrupt occurs when a message has been received and stored in message buffer k-1, indicating that MBRB has become full. Alternatively, by clearing the IE bit of message buffers 0 to (k-3) and setting the IE bit of message buffer k-2, a warning that MBRB is about to overflow can be issued.

The basic conditions of storing receive data in each message buffer for the MBRB are the same as the conditions of storing data in a single message buffer.

Cautions 1. MBRB can be configured for each of the same message buffer types. Therefore, even if a message buffer of another MBRB whose ID matches but whose message buffer type is different has a vacancy, the received message is not stored in that message buffer, but instead discarded.

2. MBRB does not have a ring buffer structure. Therefore, after a message is stored in the message buffer having the highest number in the MBRB configuration, a newly received message will not be stored in the message buffer having the lowest message buffer number.

3. MBRB operates based on the reception and storage conditions; there are no settings dedicated to MBRB, such as function enable bits. By setting the same message buffer type and ID to two or more message buffers, MBRB is automatically configured.

4. With MBRB, “matching ID” means “matching ID after mask”. Even if the ID set to each message buffer is not the same, if the ID that is masked by the mask register matches, it is considered a matching ID and the buffer that has this ID is treated as the storage destination of a message.

5. The priority between MBRBs is mentioned in 13.9.1 Message Reception.

Remark m = 0 to 15

13.9.6 Remote frame reception

In all the operation modes, when a remote frame is received, the message buffer that is to store the remote frame is searched from all the message buffers satisfying the following conditions.

- Used as a message buffer
  (MA0 bit of CMCONFm register set to 1B.)
- Set as a transmit message buffer
  (MT[2:0] bits in CMCONFm register set to 000B)
- Ready for reception
  (RDY bit of CMCTRLm register set to 1.)
- Set to transmit message
  (RTR bit of CMCONFm register is cleared to 0.)
- Transmission request is not set.
  (TRQ bit of CMCTRLm register is cleared to 0.)

Upon acceptance of a remote frame, the following actions are executed if the ID of the received remote frame matches the ID of a message buffer that satisfies the above conditions.

- The MDLC[3:0] bit string in the CMDLCm register stores the received DLC value.
- CMDATA0m to CMDATA7m in the data area are not updated (data before reception is saved).
- The DN bit of the CMCTRLm register is set to 1.
- The CINTS1 bit of the CINTS register is set to 1 (if the IE bit in the CMCTRLm register of the message buffer that receives and stores the frame is set to 1).
- The reception completion interrupt (INTCREC) is output (if the IE bit in the CMCTRLm register of the message buffer that receives and stores the frame is set to 1 and if the CIE1 bit of the CIE register is set to 1).
- The message buffer number is recorded to the receive history list.

Caution When a message buffer is searched for receiving and storing a remote frame, overwrite control by the OWS bit of the CMCONFm register of the message buffer and the DN bit of the CMCTRLm register are not affected. The setting of OWS is ignored, and DN is set in any case.
If more than one transmit message buffer has the same ID and the ID of the received remote frame matches that ID, the remote frame is stored in the transmit message buffer with the lowest message buffer number.

Remark m = 0 to 15

13.10 Message Transmission

13.10.1 Message transmission

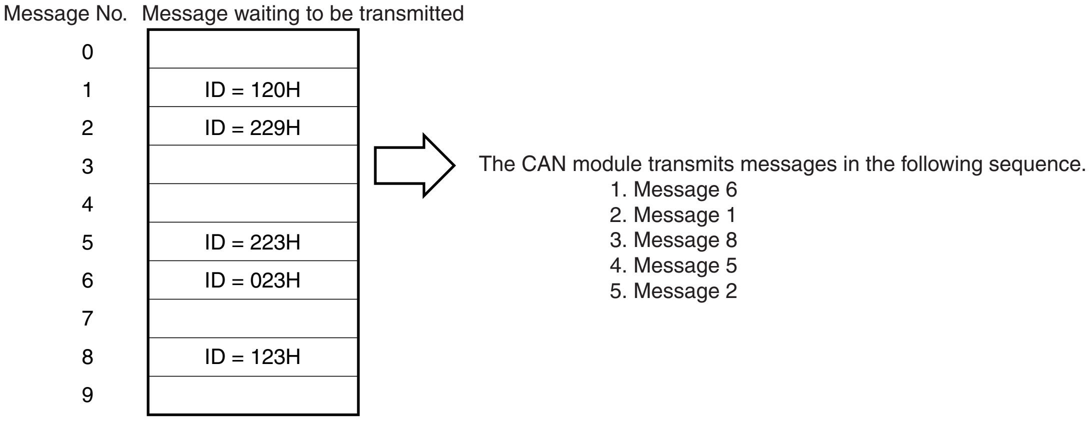
In all the operation modes, if the TRQ bit is set to 1 in a message buffer that satisfies the following conditions, the message buffer that is to transmit a message is searched.

- Used as a message buffer
  (MA0 bit of CMCONFm register set to 1B.)
- Set as a transmit message buffer
  (MT[2:0] bits of CMCONFm register set to 000B.)
- Ready for transmission
  (RDY bit of CMCTRLm register set to 1.)

The CAN system is a multi-master communication system. In a system like this, the priority of message transmission is determined based on message identifiers (IDs). To facilitate transmission processing by software when there are several messages awaiting transmission, the CAN module uses hardware to check the ID of the message with the highest priority and automatically identifies that message. This eliminates the need for software-based priority control.

Transmission priority is controlled by the identifier (ID).

Figure 13-57. Message Processing Example



Message No.	Message waiting to be transmitted
0	
1	ID = 120H
2	ID = 229H
3	
4	
5	ID = 223H
6	ID = 023H
7	
8	ID = 123H
9	

The CAN module transmits messages in the following sequence.
1. Message 6
2. Message 1
3. Message 8
4. Message 5
5. Message 2

After the transmit message search, the transmit message with the highest priority of the transmit message buffers that have a pending transmission request (message buffers with the TRQ bit set to 1 in advance) is transmitted.

If a new transmission request is set, the transmit message buffer with the new transmission request is compared with the transmit message buffer with a pending transmission request. If the new transmission request has a higher priority, it is transmitted, unless transmission of a message with a low priority has already started. If transmission of a message with a low priority has already started, however, the new transmission request is transmitted later. To solve this priority inversion effect, the software can perform a transmission abort request for the lower priority message. The highest priority is determined according to the following rules.

Priority	Conditions	Description
1(high)	Value of first 11 bits of ID [ID28 to ID18]:	The message frame with the lowest value represented by the first 11 bits of the ID is transmitted first. If the value of an 11-bit standard ID is equal to or smaller than the first 11 bits of a 29-bit extended ID, the 11-bit standard ID has a higher priority than message frame with the 29-bit extended ID.
2	Frame type	A data frame with an 11-bit standard ID (RTR bit is cleared to 0) has a higher priority than a remote frame with a standard ID and a message frame with an extended ID.
3	ID type	A message frame with a standard ID (IDE bit is cleared to 0) has a higher priority than a message frame with an extended ID.
4	Value of lower 18 bits of ID [ID17 to ID0]:	If more than one transmission-pending extended ID message frame have equal values in the first 11 bits of the ID and the same frame type (equal RTR bit values), the message frame with the lowest value in the lower 18 bits of its extended ID is transmitted first.
5(low)	Message buffer number	If two or more message buffers request transmission of message frames with the same ID, the message from the message buffer with the lowest message buffer number is transmitted first.

Remarks 1. If automatic block transmission request bit ABTTRG is set to 1 in the normal operation mode with ABT, the TRQ bit is set to 1 only for one message buffer in the ABT message buffer group.

If the ABT mode was triggered by ABTTRG bit, one TRQ bit is set to 1 in the ABT area (buffer 0 through 7). Beyond this TRQ bit, the application can request transmissions (set TRQ to 1) for other TX-message buffers that do not belong to the ABT area. In that case an interval arbitration process (TX-search) evaluates all TX-message buffers with TRQ bit set to 1 and chooses the message buffer that contains the highest prioritized identifier for the next transmission. If there are 2 or more identifiers that have the highest priority (i.e. identical identifiers), the message located at the lowest message buffer number is transmitted at first.

Upon successful transmission of a message frame, the following operations are performed.

- The TRQ flag of the corresponding transmit message buffer is automatically cleared to 0.
- The transmission completion status bit CINTS0 of the CINTS register is set to 1 (if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).
- An interrupt request signal INTC0TRX output (if the CIE0 bit of the CIE register is set to 1 and if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).

2. When changing the contents of a transmit buffer, the RDY flag of this buffer must be cleared before updating the buffer contents. As during internal transfer actions, the RDY flag may be locked temporarily, the status of RDY must be checked by software, after changing it.

3. m = 0 to 15

13.10.2 Transmit history list function

The transmit history list (THL) function records in the transmit history list the number of the transmit message buffer from which data or remote frames have been were sent. The THL consists of storage elements equivalent to up to seven messages, the last out-message pointer (LOPT) with the corresponding CLOPT register, and the transmit history list get pointer (TGPT) with the corresponding CTGPT register.

The THL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The CLOPT register holds the contents of the THL element indicated by the value of the LOPT pointer minus 1. By reading the CLOPT register, therefore, the number of the message buffer that transmitted a data frame or remote frame first can be checked. The LOPT pointer is utilized as a write pointer that indicates to what part of the THL a message buffer number is recorded. Any time a data frame or remote frame is transmitted, the corresponding message buffer number is recorded to the THL element indicated by the LOPT pointer. Each time recording to the THL has been completed, the LOPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

The TGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the THL. This pointer indicates the first THL element that the CPU has not yet read. By reading the CTGPT register by software, the number of a message buffer that has completed transmission can be read. Each time a message buffer number is read from the CTGPT register, the TGPT pointer is automatically incremented.

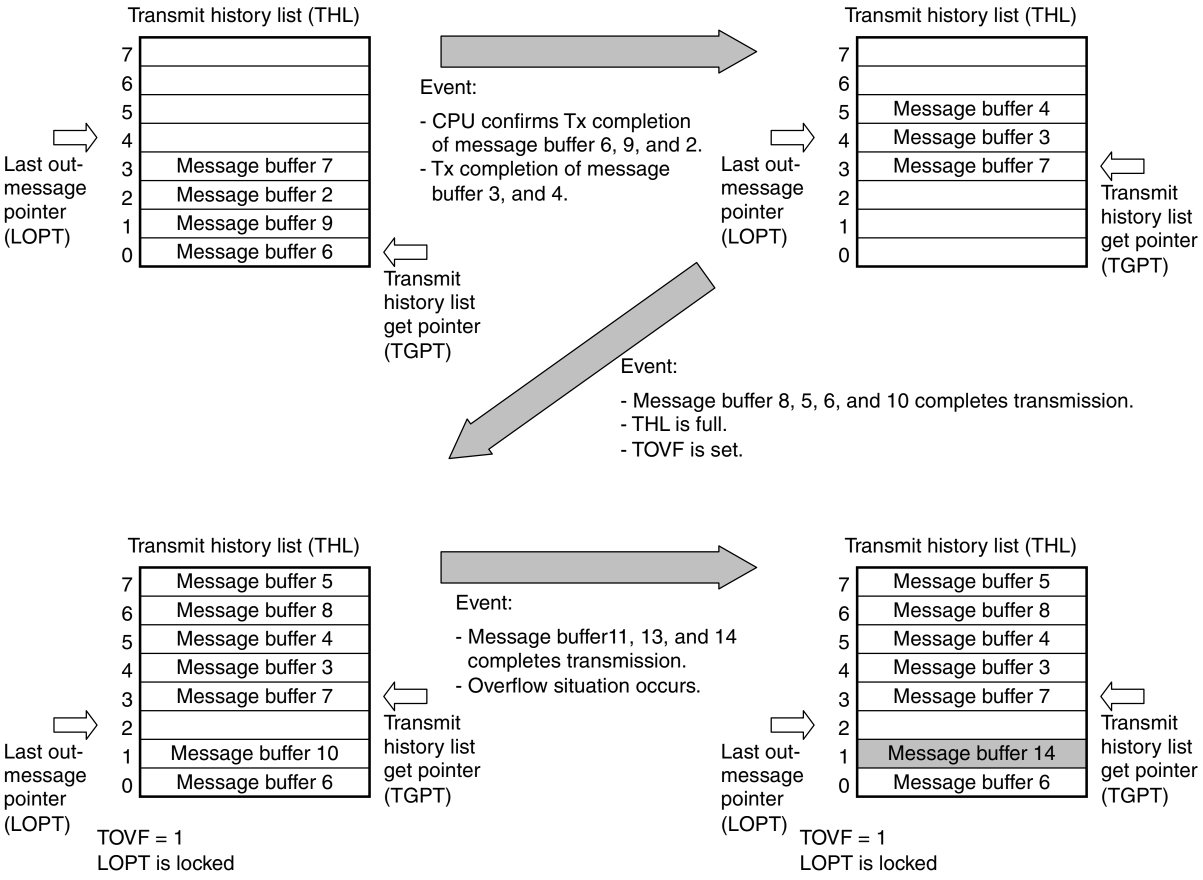
If the value of the TGPT pointer matches the value of the LOPT pointer, the THPM bit (transmit history list pointer match) of the CTGPT register is set to 1. This indicates that no message buffer numbers that have not been read remain in the THL. If a new message buffer number is recorded, the LOPT pointer is incremented and because its value no longer matches the value of the TGPT pointer, the THPM bit is cleared. In other words, the numbers of the unread message buffers exist in the THL.

If the LOPT pointer is incremented and matches the value of the TGPT pointer minus 1, the TOVF bit (transmit history list overflow) of the CTGPT register is set to 1. This indicates that the THL is full of message buffer numbers that have not been read. If a new message is received and stored, the message buffer number recorded last is overwritten by the number of the message buffer that transmitted its message afterwards. After the TOVF bit has been set (1), therefore, the recorded message buffer numbers in the THL do not completely reflect the chronological order. However the other transmitted messages can be found by a CPU search applied to all transmit message buffers unless the CPU has not overwritten a transmit object in one of these buffers beforehand. In total up to six transmission completions can occur without overflowing the THL.

Caution **If the history list is in the overflow condition (TOVF is set), reading the history list contents is still possible, until the history list is empty (indicated by THPM flag set). Nevertheless, the history list remains in the overflow condition, until TOVF is cleared by software. If TOVF is not cleared, the THPM flag will also not be updated (cleared) upon successful transmission of a new message. This may lead to the situation, that THPM indicates an empty history list, although a successful transmission has taken place, while the history list is in the overflow state (TOVF and THPM are set).**

Remark m = 0 to 15

Figure 13-58. Transmit History List


Transmit history list (THL)
Last out-message pointer (LOPT)
Transmit history list get pointer (TGPT)
TOVF = 1
LOPT is locked

Transmit history list (THL)

7	
6	
5	
4	
3	Message buffer 7
2	Message buffer 2
1	Message buffer 9
0	Message buffer 6

Event:

- CPU confirms Tx completion of message buffer 6, 9, and 2.
- Tx completion of message buffer 3, and 4.

Transmit history list (THL)

7	
6	
5	Message buffer 4
4	Message buffer 3
3	Message buffer 7
2	
1	
0	

Event:

- Message buffer 8, 5, 6, and 10 completes transmission.
- THL is full.
- TOVF is set.

Transmit history list (THL)

7	Message buffer 5
6	Message buffer 8
5	Message buffer 4
4	Message buffer 3
3	Message buffer 7
2	
1	Message buffer 10
0	Message buffer 6

Event:

- Message buffer11, 13, and 14 completes transmission.
- Overflow situation occurs.

Transmit history list (THL)

7	Message buffer 5
6	Message buffer 8
5	Message buffer 4
4	Message buffer 3
3	Message buffer 7
2	
1	Message buffer 14
0	Message buffer 6

TOVF = 1 denotes that LOPT equals TGPT−1 while message buffer number stored to element indicated by LOPT−1.

13.10.3 Automatic block transmission (ABT)

The automatic block transmission (ABT) function is used to transmit two or more data frames successively with no CPU interaction. The maximum number of transmit message buffers assigned to the ABT function is eight (message buffer numbers 0 to 7).

By setting OPMODE[2:0] bits of the CCTRL register to 010B, “normal operation mode with automatic block transmission function” (hereafter referred to as ABT mode) can be selected.

To issue an ABT transmission request, define the message buffers by software first. Set the MA0 bit (1) in all the message buffers used for ABT, and define all the buffers as transmit message buffers by setting MT[2:0] bits to 000B. Be sure to set the ID for each message buffer for ABT even when the same ID is being used for all the message buffers. To use two or more IDs, set the ID of each message buffer by using the CMIDLm and CMIDHm registers. Set the CMDLCm and CMDATA0m to CMDATA7m registers before issuing a transmission request for the ABT function.

After initialization of message buffers for ABT is finished, the RDY bit needs to be set (1). In the ABT mode, the TRQ bit does not have to be manipulated by software.

After the data for the ABT message buffers has been prepared, set the ABTTRG bit to 1. Automatic block transmission is then started. When ABT is started, the TRQ bit in the first message buffer (message buffer 0) is automatically set to 1. After transmission of the data of message buffer 0 has finished, TRQ bit of the next message buffer, message buffer 1, is set automatically. In this way, transmission is executed successively.

A delay time can be inserted by program in the interval in which the transmission request (TRQ) is automatically set while successive transmission is being executed. The delay time to be inserted is defined by the CGMABTD register. The unit of the delay time is DBT (data bit time). DBT depends on the setting of the CBRP and CBTR registers.

Among transmit objects within the ABT-area, the priority of the transmission ID is not evaluated. The data of message buffers 0 to 7 are sequentially transmitted. When transmission of the data frame from message buffer 7 has been completed, the ABTTRG bit is automatically cleared to 0 and the ABT operation is finished.

If the RDY bit of an ABT message buffer is cleared during ABT, no data frame is transmitted from that buffer, ABT is stopped, and the ABTTRG bit is cleared. After that, transmission can be resumed from the message buffer where ABT stopped, by setting the RDY and ABTTRG bits to 1 by software. To not resume transmission from the message buffer where ABT stopped, the internal ABT engine can be reset by setting the ABTCLR bit to 1 while ABT mode is stopped and ABTTRG bit is cleared to 0. In this case, transmission is started from message buffer 0 if the ABTCLR bit is cleared to 0 and then the ABTTRG bit is set to 1.

An interrupt can be used to check if data frames have been transmitted from all the message buffers for ABT. To do so, the IE bit of the CMCTRLm register of each message buffer except the last message buffer needs to be cleared (0).

If a transmit message buffer other than those used by the ABT function (message buffer 8 to 15) is assigned to a transmit message buffer, the message to be transmitted next is determined by the priority of the transmission ID of the ABT message buffer whose transmission is currently held pending and the transmission ID of the message buffers other than those used by the ABT function.

Transmission of a data frame from an ABT message buffer is not recorded in the transmit history list (THL).

Cautions 1. Set the ABTCLR bit to 1 while the ABTTRG bit is cleared to 0 in order to resume ABT operation at buffer No.0. If the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1, the subsequent operation is not guaranteed.

2. If the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared immediately after the processing of the clearing request is completed.

3. Do not set the ABTTRG bit in the initialization mode. If the ABTTRG bit is set in the initialization mode, the proper operation is not guaranteed after the mode is changed from the initialization mode to the ABT mode.

4. Do not set TRQ bit of the ABT message buffers to 1 by software in the normal operation mode with ABT. Otherwise, the operation is not guaranteed.

5. The CGMABTD register is used to set the delay time that is inserted in the period from completion of the preceding ABT message to setting of the TRQ bit for the next ABT message when the transmission requests are set in the order of message numbers for each message for ABT that is successively transmitted in the ABT mode. The timing at which the messages are actually transmitted onto the CAN bus varies depending on the status of transmission from other stations and the status of the setting of the transmission request for messages other than the ABT messages (message buffer 8 to 15).

6. If a transmission request is made for a message other than an ABT message and if no delay time is inserted in the interval in which transmission requests for ABT are automatically set (CGMABTD = 00H), messages other than ABT messages may be transmitted not depending on the priority of the ABT message.

7. Do not clear the RDY bit to 0 when ABTTRG = 1.

8. If a message is received from another node while normal operation mode with ABT is active, the TX-message from the ABT-area may be transmitted with delay of one frame although CGMABTD register was set up with 00H.

Remark m = 0 to 15

13.10.4 Transmission abort process

(1) Transmission abort process except for in normal operation mode with automatic block transmission (ABT)

The user can clear the TRQ bit of the CMCTRLm register to 0 to abort a transmission request. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the CCTRL register and the CTGPT register, which indicate the transmission status on the CAN bus (for details, see the processing in **Figure 13-72**).

(2) Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)

The user can clear the ABTTRG bit of the CGMABT register to 0 to abort a transmission request. After checking the ABTTRG bit of the CGMABT register = 0, clear the TRQ bit of the CMCTRLm register to 0. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the CCTRL register and the CTGPT register, which indicate the transmission status on the CAN bus (for details, see the processing in **Figure 13-74**).

(3) Transmission abort process for ABT transmission in normal operation mode with automatic block transmission (ABT)

To abort ABT that is already started, clear the ABTTRG bit of the CGMABT register to 0. In this case, the ABTTRG bit remains 1 if an ABT message is currently being transmitted and until the transmission is completed (successfully or not), and is cleared to 0 as soon as transmission is finished. This aborts ABT.

If the last transmission (before ABT) was successful, the normal operation mode with ABT is left with the internal ABT pointer pointing to the next message buffer to be transmitted.

In the case of an erroneous transmission, the position of the internal ABT pointer depends on the status of the TRQ bit in the last transmitted message buffer. If the TRQ bit is set to 1 when clearing the ABTTRG bit is requested, the internal ABT pointer points to the last transmitted message buffer (for details, see the process in **Figure 13-73**). If the TRQ bit is cleared to 0 when clearing the ABTTRG bit is requested, the internal ABT pointer is incremented (+1) and points to the next message buffer in the ABT area (for details, see the process in **Figure 13-74**).

Caution Be sure to abort ABT by clearing ABTTRG to 0. The operation is not guaranteed if aborting transmission is requested by clearing RDY bit.

When the normal operation mode with ABT is resumed after ABT has been aborted and ABTTRG bit is set to 1, the next ABT message buffer to be transmitted can be determined from the following table.

Status of TRQ of ABT Message Buffer	Abort After Successful Transmission	Abort After Erroneous Transmission
Set (1)	Next message buffer in the ABT area[Note]	Same message buffer in the ABT area
Cleared (0)	Next message buffer in the ABT area[Note]	Next message buffer in the ABT area[Note]

Note The above resumption operation can be performed only if a message buffer ready for ABT exists in the ABT area. For example, an abort request that is issued while ABT of message buffer 7 is in progress is regarded as completion of ABT, rather than abort, if transmission of message buffer 7 has been successfully completed, even if ABTTRG is cleared to 0. If the RDY bit in the next message buffer in the ABT area is cleared to 0, the internal ABT pointer is retained, but the resumption operation is not performed even if ABTTRG is set to 1, and ABT ends immediately.

Remark m = 0 to 15

13.10.5 Remote frame transmission

Remote frames can be transmitted only from transmit message buffers. Set whether a data frame or remote frame is transmitted via the RTR bit of the CMCONFm register. Setting (1) the RTR bit sets remote frame transmission.

Remark m = 0 to 15

13.11 Power Save Modes

13.11.1 CAN sleep mode

The CAN sleep mode can be used to set the CAN controller to standby mode in order to reduce power consumption. The CAN module can enter the CAN sleep mode from all operation modes. Release of the CAN sleep mode returns the CAN module to exactly the same operation mode from which the CAN sleep mode was entered.

In the CAN sleep mode, the CAN module does not transmit messages, even when transmission requests are issued or pending.

(1) Entering CAN sleep mode

The CPU issues a CAN sleep mode transition request by writing 01B to the PSMODE[1:0] bits of the CCTRL register.

This transition request is only acknowledged only under the following conditions.

(I) The CAN module is already in one of the following operation modes

- Normal operation mode
- Normal operation mode with ABT
- Receive-only mode
- Single-shot mode
- Self-test mode
- CAN stop mode in all the above operation modes

(II) The CAN bus state is bus idle (the 4th bit in the interframe space is recessive)[Note]

(III) No transmission request is pending

Note If the CAN bus is fixed to dominant, the request for transition to the CAN sleep mode is held pending. Also the transition from CAN stop mode to CAN sleep mode is independent of the CAN bus state.

Remark If a sleep mode request is pending, and at the same time a message is received in a message box, the sleep mode request is not cancelled, but is executed right after message storage has been finished. This may result in AFCAN being in sleep mode, while the CPU would execute the RX interrupt routine. Therefore, the interrupt routine must check the access to the message buffers as well as reception history list registers by using the MBON flag, if sleep mode is used.

If any one of the conditions mentioned above is not met, the CAN module will operate as follows.

- If the CAN sleep mode is requested from the initialization mode, the CAN sleep mode transition request is ignored and the CAN module remains in the initialization mode.
- If the CAN bus state is not bus idle (i.e., the CAN bus state is either transmitting or receiving) when the CAN sleep mode is requested in one of the operation modes, immediate transition to the CAN sleep mode is not possible. In this case, the CAN sleep mode transition request is held pending until the CAN bus state becomes bus idle (the 4th bit in the interframe space is recessive). In the time from the CAN sleep mode request to successful transition, the PSMODE[1:0] bits remain 00B. When the module has entered the CAN sleep mode, PSMODE[1:0] bits are set to 01B.
- If a request for transition to the initialization mode and a request for transition to the CAN sleep are made at the same time while the CAN module is in one of the operation modes, the request for the initialization mode is enabled. The CAN module enters the initialization mode at a predetermined timing. At this time, the CAN sleep mode request is not held pending and is ignored.

- Even when initialization mode and sleep mode are not requested simultaneously (i.e the first request has not been granted while the second request is made), the request for initialization has priority over the sleep mode request. The sleep mode request is cancelled when the initialization mode is requested. When a pending request for initialization mode is present, a subsequent request for Sleep mode request is cancelled right at the point in time where it was submitted.

(2) Status in CAN sleep mode

The CAN module is in one of the following states after it enters the CAN sleep mode.

- The internal operating clock is stopped and the power consumption is minimized.
- The function to detect the falling edge of the CAN reception pin (CRxD) remains in effect to wake up the CAN module from the CAN bus.
- To wake up the CAN module from the CPU, data can be written to PSMODE[1:0] of the CAN module control register (CCTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for CLIPT, CRGPT, CLOPT, and CTGPT.
- The CAN message buffer registers cannot be written or read.
- MBON bit of the CAN Global Control register (CGMCTRL) is cleared.
- A request for transition to the initialization mode is not acknowledged and is ignored.

(3) Releasing CAN sleep mode

The CAN sleep mode is released by the following events.

- When the CPU writes 00B to the PSMODE[1:0] bits of the CCTRL register
- A falling edge at the CAN reception pin (CRxD) (i.e. the CAN bus level shifts from recessive to dominant)

Cautions 1. Even if the falling edge belongs to the SOF of a receive message, this message will not be received and stored. If the CPU has turned off the clock to the CAN while the CAN was in sleep mode, even subsequently the CAN sleep mode will not be released and PSMODE[1:0] will continue to be 01B unless the clock to the CAN is supplied again. In addition to this, the receive message will not be received after that.

2. If the falling edge on the CAN reception pin (CRxD) is detected in the state that the CAN clock is supplied, it is necessary to clear the PSMODE0 bit by software (for details, see the processing in Figure 13-81).

After releasing the sleep mode, the CAN module returns to the operation mode from which the CAN sleep mode was requested and the PSMODE[1:0] bits of the CCTRL register are reset to 00B. If the CAN sleep mode is released by a change in the CAN bus state, the CINTS5 bit of the CINTS register is set to 1, regardless of the CIE bit of the CIE register. After the CAN module is released from the CAN sleep mode, it participates in the CAN bus again by automatically detecting 11 consecutive recessive-level bits on the CAN bus. The user application has to wait until MBON = 1, before accessing message buffers again.

When a request for transition to the initialization mode is made while the CAN module is in the CAN sleep mode, that request is ignored; the CPU has to be released from sleep mode by software first before entering the initialization mode.

Caution Be aware that the release of CAN sleep mode by CAN bus event, and thus the wake up interrupt may happen at any time, even right after requesting sleep mode, if a CAN bus event occurs.

Remark m = 0 to 15

13.11.2 CAN stop mode

The CAN stop mode can be used to set the CAN controller to standby mode to reduce power consumption. The CAN module can enter the CAN stop mode only from the CAN sleep mode. Release of the CAN stop mode puts the CAN module in the CAN sleep mode.

The CAN stop mode can only be released (entering CAN sleep mode) by writing 01B to the PSMODE[1:0] bits of the CCTRL register and not by a change in the CAN bus state. No message is transmitted even when transmission requests are issued or pending.

(1) Entering CAN stop mode

A CAN stop mode transition request is issued by writing 11B to the PSMODE[1:0] bits of the CCTRL register.

A CAN stop mode request is only acknowledged when the CAN module is in the CAN sleep mode. In all other modes, the request is ignored.

Caution To set the CAN module to the CAN stop mode, the module must be in the CAN sleep mode. To confirm that the module is in the sleep mode, check that PSMODE[1:0] = 01B, and then request the CAN stop mode. If a bus change occurs at the CAN reception pin (CRxD) while this process is being performed, the CAN sleep mode is automatically released. In this case, the CAN stop mode transition request cannot be acknowledged (However, in the state that the CAN clock is supplied, it is necessary to clear the PSMODE0 bit by software after a bus change occurs at the CAN reception pin (CRxD)).

(2) Status in CAN stop mode

The CAN module is in one of the following states after it enters the CAN stop mode.

- The internal operating clock is stopped and the power consumption is minimized.
- To wake up the CAN module from the CPU, data can be written to PSMODE[1:0] of the CAN module control register (CCTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for CLIPT, CRGPT, CLOPT, and CTGPT.
- The CAN message buffer registers cannot be written or read.
- MBON bit of the CAN Global Control register (CGMCTRL) is cleared.
- An initialization mode transition request is not acknowledged and is ignored.

(3) Releasing CAN stop mode

The CAN stop mode can only be released by writing 01B to the PSMODE[1:0] bits of the CCTRL register. After releasing the CAN stop mode, the CAN module enters the CAN sleep mode.

When the initialization mode is requested while the CAN module is in the CAN stop mode, that request is ignored; the CPU has to release the stop mode and subsequently CAN sleep mode before entering the initialization mode. It is impossible to enter the other operation mode directly from the CAN stop mode not entering the CAN sleep mode, that request is ignored.

Remark m = 0 to 15

<R> 13.11.3 Example of using power saving modes

In some application systems, it may be necessary to place the CPU in a power saving mode to reduce the power consumption. By using the power saving mode specific to the CAN module and the power saving mode specific to the CPU in combination, the CPU can be woken up from the power saving status by the CAN bus.

Here is an example of using the power saving modes.

First, put the CAN module in the CAN sleep mode (PSMODE = 01B).

Next, CPU is made to change to the power save mode which stops the clock supply to CPU.

When a CAN receiving pin (CRxD) detects the edge change to dominant from recessive in the state, the CAN module, It is possible to generate the wakeup interruption (INTC0WUP) with a setup of a CINTS5 bit(=1), when the CIE5 bit of the CCTRL register is 1 further also after the clock has stopped.

The CPU, in response to INTC0WUP, releases its power saving mode, resumes supply of the internal clocks, including the clock to the CAN module, after the oscillation stabilization time has elapsed, and starts instruction execution. The CAN module is immediately released from the CAN sleep mode when clock supply is resumed, and returns to the normal operation mode (PSMODE = 00B).

13.12 Interrupt Function

The CAN module provides 6 different interrupt sources.

The occurrence of these interrupt sources is stored in interrupt status registers. Four separate interrupt request signals are generated from the six interrupt sources. When an interrupt request signal that corresponds to two or more interrupt sources is generated, the interrupt sources can be identified by using an interrupt status register. After an interrupt source has occurred, the corresponding interrupt status bit must be cleared to 0 by software.

Table 13-20. List of CAN Module Interrupt Sources

No.	Interrupt Status Bit		Interrupt Enable Bit		Interrupt Request Signal	Interrupt Source Description
	Name	Register	Name	Register		
1	CINTS0[Note]	CINTS	CIE0[Note]	CIE	INTC0TRX	Message frame successfully transmitted from message buffer m
2	CINTS1[Note]	CINTS	CIE1[Note]	CIE	INTC0REC	Valid message frame reception in message buffer m
3	CINTS2	CINTS	CIE2	CIE	INTC0ERR	CAN module error state interrupt (Supplement 1)
4	CINTS3	CINTS	CIE3	CIE		CAN module protocol error interrupt (Supplement 2)
5	CINTS4	CINTS	CIE4	CIE		CAN module arbitration loss interrupt
6	CINTS5	CINTS	CIE5	CIE	INTC0WUP	CAN module wakeup interrupt from CAN sleep mode (Supplement 3)

Note The IE bit (message buffer interrupt enable bit) in the CMCTRL register of the corresponding message buffer has to be set to 1 for that message buffer to participate in the interrupt generation process.

Supplements 1. This interrupt is generated when the transmission/reception error counter is at the warning level, or in the error passive or bus-off state.

2. This interrupt is generated when a stuff error, form error, ACK error, bit error, or CRC error occurs.

3. This interrupt is generated when the CAN module is woken up from the CAN sleep mode because a falling edge is detected at the CAN reception pin (CAN bus transition from recessive to dominant).

Remark m = 0 to 15

13.13 Diagnosis Functions and Special Operational Modes

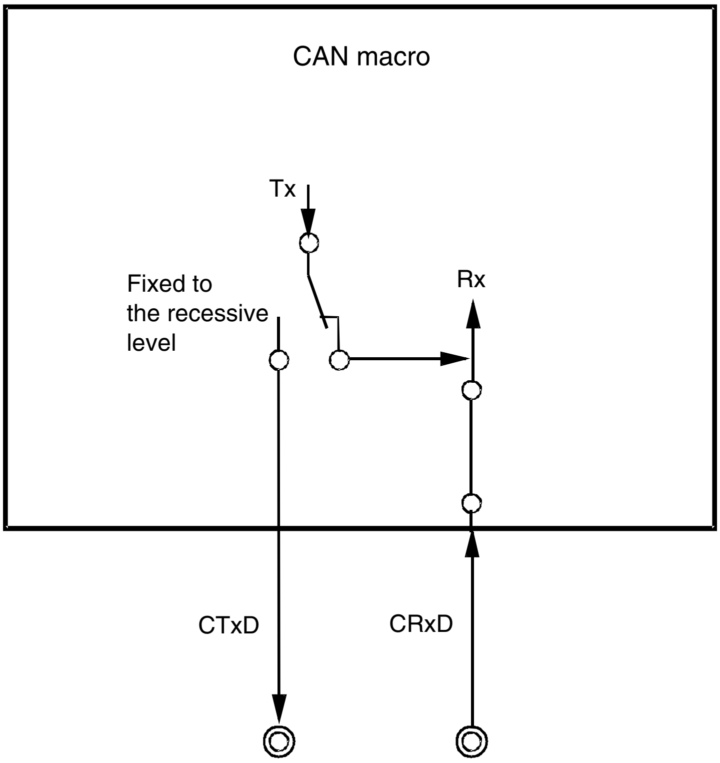
The CAN module provides a receive-only mode, single-shot mode, and self-test mode to support CAN bus diagnosis functions or the operation of specific CAN communication methods.

13.13.1 Receive-only mode

The receive-only mode is used to monitor receive messages without causing any interference on the CAN bus and can be used for CAN bus analysis nodes.

For example, this mode can be used for automatic baud-rate detection. The baud rate in the CAN module is changed until “valid reception” is detected, so that the baud rates in the module match (“valid reception” means a message frame has been received in the CAN protocol layer without occurrence of an error and with an appropriate ACK between nodes connected to the CAN bus). A valid reception does not require message frames to be stored in a receive message buffer (data frames) or transmit message buffer (remote frames). The event of valid reception is indicated by setting the VALID bit of the CCTRL register (1).

Figure 13-59. CAN Module Terminal Connection in Receive-Only Mode



In the receive-only mode, no message frames can be transmitted from the CAN module to the CAN bus. Transmit requests issued for message buffers defined as transmit message buffers are held pending.

In the receive-only mode, the CAN transmission pin (CTxD) in the CAN module is fixed to the recessive level. Therefore, no active error flag can be transmitted from the CAN module to the CAN bus even when a CAN bus error is detected while receiving a message frame. Since no transmission can be issued from the CAN module, the transmission error counter TEC is never updated. Therefore, a CAN module in the receive-only mode does not enter the bus-off state.

Furthermore, ACK is not returned to the CAN bus in this mode upon the valid reception of a message frame. Internally, the local node recognizes that it has transmitted ACK. An overload frame cannot be transmitted to the CAN bus.

Caution If only two CAN nodes are connected to the CAN bus and one of them is operating in the receive-only mode, there is no ACK on the CAN bus. Due to the missing ACK, the transmitting node will transmit an active error flag, and repeat transmitting a message frame. The transmitting node becomes error passive after transmitting the message frame 16 times (assuming that the error counter was 0 in the beginning and no other errors have occurred). After the message frame for the 17th time is transmitted, the transmitting node generates a passive error flag. The receiving node in the receive-only mode detects the first valid message frame at this point, and the VALID bit is set to 1 for the first time.

13.13.2 Single-shot mode

In the single-shot mode, automatic re-transmission as defined in the CAN protocol is switched off (According to the CAN protocol, a message frame transmission that has been aborted by either arbitration loss or error occurrence has to be repeated without control by software.). All other behavior of single shot mode is identical to normal operation mode. Features of single shot mode can not be used in combination with normal mode with ABT.

The single-shot mode disables the re-transmission of an aborted message frame transmission according to the setting of the AL bit of the CCTRL register. When the AL bit is cleared to 0, re-transmission upon arbitration loss and upon error occurrence is disabled. If the AL bit is set to 1, re-transmission upon error occurrence is disabled, but re-transmission upon arbitration loss is enabled. As a consequence, the TRQ bit in a message buffer defined as a transmit message buffer is cleared to 0 by the following events.

- Successful transmission of the message frame
- Arbitration loss while sending the message frame
- Error occurrence while sending the message frame

The events arbitration loss and error occurrence can be distinguished by checking the CINTS4 and CINTS3 bits of the CINTS register respectively, and the type of the error can be identified by reading the LEC[2:0] bits of the CLEC register.

Upon successful transmission of the message frame, the transmit completion interrupt bit CINTS0 of the CINTS register is set to 1. If the CIE0 bit of the CIE register is set to 1 at this time, an interrupt request signal is output.

The single-shot mode can be used when emulating time-triggered communication methods (e.g. TTCAN level 1).

Caution The AL bit is only valid in Single-shot mode. It does not influence the operation of re-transmission upon arbitration loss in the other operation modes.

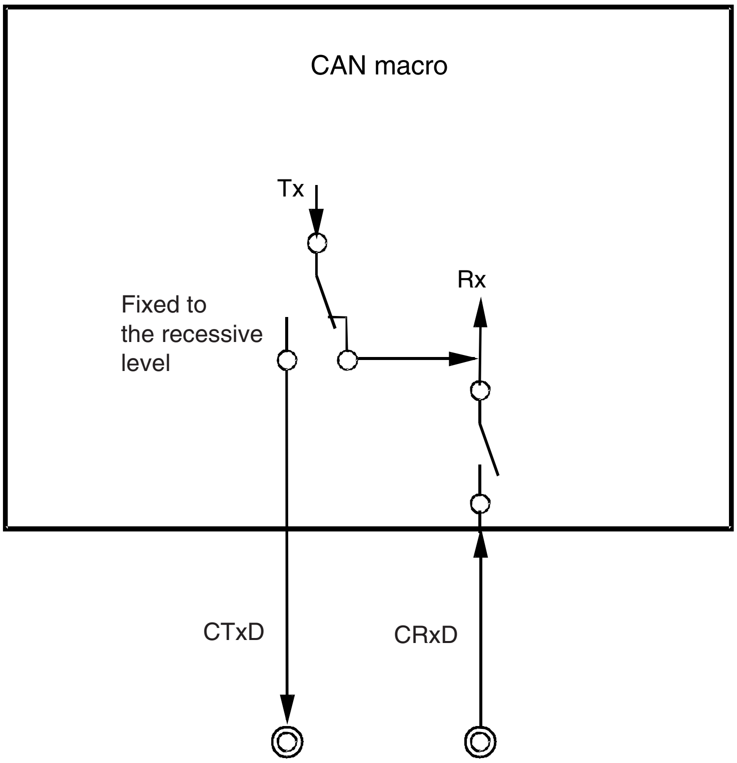
13.13.3 Self-test mode

In the self-test mode, message frame transmission and message frame reception can be tested without connecting the CAN node to the CAN bus or without affecting the CAN bus.

In the self-test mode, the CAN module is completely disconnected from the CAN bus, but transmission and reception are internally looped back. The CAN transmission pin (CTxD) is fixed to the recessive level.

If the falling edge on the CAN reception pin (CRxD) is detected after the CAN module has entered the CAN sleep mode from the self-test mode, however, the module is released from the CAN sleep mode in the same manner as the other operation modes (However, to release the CAN sleep mode in the state that the CAN clock is supplied, it is necessary to clear the PSMODE0 bit by software after the falling edge on the CAN reception pin (CRxD) is detected). To keep the module in the CAN sleep mode, use the CAN reception pin (CRxD) as a port pin.

Figure 13-60. CAN Module Terminal Connection in Self-test Mode



13.13.4 Receive/transmit operation in each operation mode

Table 13-21 shows outline of the receive/transmit operation in each operation mode.

Table 13-21. Outline of the Receive/Transmit in Each Operation Mode

Operation Mode	Transmission of Data/ RemoteFrame	Transmission of ACK	Transmission of Error/ Overload frame	Transmission Retry	Automatic Block Transmission (ABT)	Set of VALID Bit	Store Data to Message Buffer
Initialization Mode	No	No	No	No	No	No	No
Normal Operation Mode	Yes	Yes	Yes	Yes	No	Yes	Yes
Normal Operation Mode with ABT	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Receive-only mode	No	No	No	No	No	Yes	Yes
Single-shot Mode	Yes	Yes	Yes	No[Note 1]	No	Yes	Yes
Self-test Mode	Yes[Note 2]	Yes[Note 2]	Yes[Note 2]	Yes[Note 2]	No	Yes[Note 2]	Yes[Note 2]

Notes 1. When the arbitration lost occurs, control of re-transmission is possible by the AL bit of CCTRL register.

2. Each signals are not generated to outside, but generated into the CAN module.

13.14 Time Stamp Function

CAN is an asynchronous, serial protocol. All nodes connected to the CAN bus have a local, autonomous clock. As a consequence, the clocks of the nodes have no relation (i.e., the clocks are asynchronous and may have different frequencies).

In some applications, however, a common time base over the network (= global time base) is needed. In order to build up a global time base, a time stamp function is used. The essential mechanism of a time stamp function is the capture of timer values triggered by signals on the CAN bus.

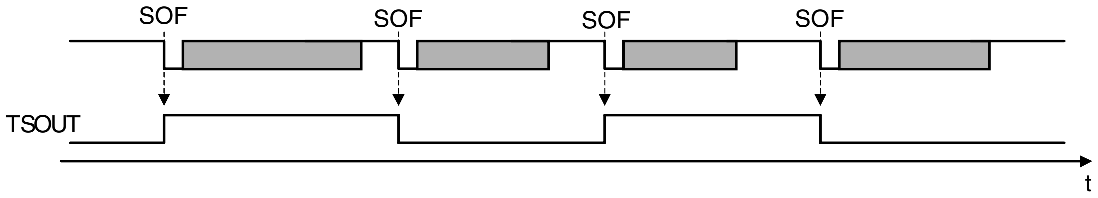
13.14.1 Time stamp function

The CAN controller supports the capturing of timer values triggered by a specific frame. An on-chip 16-bit capture timer unit in a microcontroller system is used in addition to the CAN controller. The 16-bit capture timer unit captures the timer value according to a trigger signal (TSOUT) for capturing that is output when a data frame is received from the CAN controller. The CPU can retrieve the time of occurrence of the capture event, i.e., the time stamp of the message received from the CAN bus, by reading the captured value. TSOUT signal can be selected from the following two event sources and is specified by the TSSEL bit of the CTS register.

- SOF event (start of frame)	(TSSEL = 0)
- EOF event (last bit of end of frame)	(TSSEL = 1)

The TSOUT signal is enabled by setting the TSEN bit of the CTS register to 1.

Figure 13-61. Timing Diagram of Capture Signal TSOUT



TSOUT signal toggles its level upon occurrence of the selected event during data frame reception (in the above timing diagram, the SOF is used as the trigger event source). To capture a timer value by using TSOUT signal, the capture timer unit must detect the capture signal at both the rising edge and falling edge.

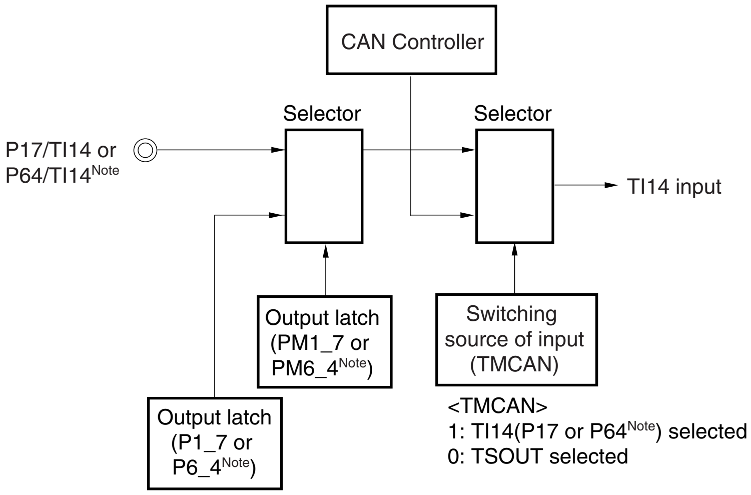
This time stamp function is controlled by the TSLOCK bit of the CTS register. When TSLOCK is cleared to 0, TSOUT bit toggles upon occurrence of the selected event. If TSLOCK bit is set to 1, TSOUT toggles upon occurrence of the selected event, but the toggle is stopped as the TSEN bit is automatically cleared to 0 as soon as the message storing to the message buffer 0 starts. This suppresses the subsequent toggle occurrence by TSOUT, so that the time stamp value toggled last (= captured last) can be saved as the time stamp value of the time at which the data frame was received in message buffer 0.

Caution **The time stamp function using TSLOCK bit is to stop toggle of TSOUT bit by receiving a data frame in message buffer 0. Therefore, message buffer 0 must be set as a receive message buffer. Since a receive message buffer cannot receive a remote frame, toggle of TSOUT bit cannot be stopped by reception of a remote frame. Toggle of TSOUT bit does not stop when a data frame is received in a message buffer other than message buffer 0.**

For these reasons, a data frame cannot be received in message buffer 0 when the CAN module is in the normal operation mode with ABT, because message buffer 0 must be set as a transmit message buffer. In this operation mode, therefore, the function to stop toggle of TSOUT bit by TSLOCK bit cannot be used.

By switching the input source (by using TMCAN), the capture trigger signal (TSOUT) can be input to channel 4 of timer array unit 1 without connecting TSOUT and TI14 externally.

Figure 13-62. Switching source of input



Note Selected by using the TIS1_4 bit.

Remarks 1. TMCAN: Bit 6 of the serial communication pin selection register (STSEL) (see **Figure 13-51**).
TIS1_4: Bit 4 of timer input selection register 1 (TIS1) (see **CHAPTER 6 TIMER ARRAY UNIT**).

2. The available pins differ depending on the product. For details, see **1.4 Pin Configuration (Top View)** and **2.1 Pin Function List**.

13.15 Baud Rate Settings

13.15.1 Baud rate settings

Make sure that the settings are within the range of limit values for ensuring correct operation of the CAN controller, as follows.

(a) 5TQ ≤ SPT (sampling point) ≤ 17TQ
SPT = TSEG1 + 1TQ
(b) 8TQ ≤ DBT (data bit time) ≤ 25TQ
DBT = TSEG1 + TSEG2 + 1TQ = TSEG2 + SPT
(c) 1TQ ≤ SJW (synchronization jump width) ≤ 4TQ
SJW ≤ DBT – SPT
(d) 4TQ ≤ TSEG1 ≤ 16TQ [3 (Setting value of TSEG1[3:0] ≤ 15]
(e) 1TQ ≤ TSEG2 ≤ 8TQ [0 (Setting value of TSEG2[2:0] ≤ 7]

Remark TQ = 1/f_{TQ} (f_{TQ}: CAN protocol layer basic system clock)
TSEG1[3:0]: Bits 3 to 0 of CAN bit rate register (CBTR)
TSEG2[2:0]: Bits 10 to 8 of CAN bit rate register (CBTR)

Table 13-22 shows the combinations of bit rates that satisfy the above conditions.

Table 13-22. Settable Bit Rate Combinations (1/3)

Valid Bit Rate Setting					CBTR Register Setting Value		Sampling Point (Unit %)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1[3:0]	TSEG2[2:0]	
25	1	8	8	8	1111	111	68.0
24	1	7	8	8	1110	111	66.7
24	1	9	7	7	1111	110	70.8
23	1	6	8	8	1101	111	65.2
23	1	8	7	7	1110	110	69.6
23	1	10	6	6	1111	101	73.9
22	1	5	8	8	1100	111	63.6
22	1	7	7	7	1101	110	68.2
22	1	9	6	6	1110	101	72.7
22	1	11	5	5	1111	100	77.3
21	1	4	8	8	1011	111	61.9
21	1	6	7	7	1100	110	66.7
21	1	8	6	6	1101	101	71.4
21	1	10	5	5	1110	100	76.2
21	1	12	4	4	1111	011	81.0
20	1	3	8	8	1010	111	60.0
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4

Table 13-22. Settable Bit Rate Combinations (2/3)

Valid Bit Rate Setting					CBTR Register Setting Value		Sampling Point (Unit %)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1[3:0]	TSEG2[2:0]	
17	1	2	7	7	1000	110	58.8
17	1	4	6	6	1001	101	64.7
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7
15	1	12	1	1	1100	000	93.3
14	1	1	6	6	0110	101	57.1
14	1	3	5	5	0111	100	64.3
14	1	5	4	4	1000	011	71.4
14	1	7	3	3	1001	010	78.6
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7

Table 13-22. Settable Bit Rate Combinations (3/3)

Valid Bit Rate Setting					CBTR Register Setting Value		Sampling Point (Unit %)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1[3:0]	TSEG2[2:0]	
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9
10	1	1	4	4	0100	011	60.0
10	1	3	3	3	0101	010	70.0
10	1	5	2	2	0110	001	80.0
10	1	7	1	1	0111	000	90.0
9	1	2	3	3	0100	010	66.7
9	1	4	2	2	0101	001	77.8
9	1	6	1	1	0110	000	88.9
8	1	1	3	3	0011	010	62.5
8	1	3	2	2	0100	001	75.0
8	1	5	1	1	0101	000	87.5
7[Note]	1	2	2	2	0011	001	71.4
7[Note]	1	4	1	1	0100	000	85.7
6[Note]	1	1	2	2	0010	001	66.7
6[Note]	1	3	1	1	0011	000	83.3
5[Note]	1	2	1	1	0010	000	80.0
4[Note]	1	1	1	1	0001	000	75.0

Note Setting with a DBT value of 7 or less is valid only when the value of the CBRP register is other than 00H.

Caution The values in Table 13-22 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

13.15.2 Representative examples of baud rate settings

Tables 13-23 and **13-24** show representative examples of baud rate setting.

Table 13-23. Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (1/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of CBRP	CBRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					CBTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	8	1	1	3	3	0011	010	62.5
1000	1	00000000	8	1	3	2	2	0100	001	75.0
1000	1	00000000	8	1	5	1	1	0101	000	87.5
500	1	00000000	16	1	1	7	7	0111	110	56.3
500	1	00000000	16	1	3	6	6	1000	101	62.5
500	1	00000000	16	1	5	5	5	1001	100	68.8
500	1	00000000	16	1	7	4	4	1010	011	75.0
500	1	00000000	16	1	9	3	3	1011	010	81.3
500	1	00000000	16	1	11	2	2	1100	001	87.5
500	1	00000000	16	1	13	1	1	1101	000	93.8
500	2	00000001	8	1	1	3	3	0011	010	62.5
500	2	00000001	8	1	3	2	2	0100	001	75.0
500	2	00000001	8	1	5	1	1	0101	000	87.5
250	2	00000001	16	1	1	7	7	0111	110	56.3
250	2	00000001	16	1	3	6	6	1000	101	62.5
250	2	00000001	16	1	5	5	5	1001	100	68.8
250	2	00000001	16	1	7	4	4	1010	011	75.0
250	2	00000001	16	1	9	3	3	1011	010	81.3
250	2	00000001	16	1	11	2	2	1100	001	87.5
250	2	00000001	16	1	13	1	1	1101	000	93.8
250	4	00000011	8	1	3	2	2	0100	001	75.0
250	4	00000011	8	1	5	1	1	0101	000	87.5
125	4	00000011	16	1	1	7	7	0111	110	56.3
125	4	00000011	16	1	3	6	6	1000	101	62.5
125	4	00000011	16	1	5	5	5	1001	100	68.8
125	4	00000011	16	1	7	4	4	1010	011	75.0
125	4	00000011	16	1	9	3	3	1011	010	81.3
125	4	00000011	16	1	11	2	2	1100	001	87.5
125	4	00000011	16	1	13	1	1	1101	000	93.8
125	8	00000111	8	1	3	2	2	0100	001	75.0
125	8	00000111	8	1	5	1	1	0101	000	87.5

Caution The values in Table 13-23 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Table 13-23. Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (2/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of CBRP	CBRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					CBTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
100	4	00000011	20	1	7	6	6	1100	101	70.0
100	4	00000011	20	1	9	5	5	1101	100	75.0
100	5	00000100	16	1	7	4	4	1010	011	75.0
100	5	00000100	16	1	9	3	3	1011	010	81.3
100	8	00000111	10	1	3	3	3	0101	010	70.0
100	8	00000111	10	1	5	2	2	0110	001	80.0
100	10	00001001	8	1	3	2	2	0100	001	75.0
100	10	00001001	8	1	5	1	1	0101	000	87.5
83.3	4	00000011	24	1	7	8	8	1110	111	66.7
83.3	4	00000011	24	1	9	7	7	1111	110	70.8
83.3	6	00000101	16	1	5	5	5	1001	100	68.8
83.3	6	00000101	16	1	7	4	4	1010	011	75.0
83.3	6	00000101	16	1	9	3	3	1011	010	81.3
83.3	6	00000101	16	1	11	2	2	1100	001	87.5
83.3	8	00000111	12	1	5	3	3	0111	010	75.0
83.3	8	00000111	12	1	7	2	2	1000	001	83.3
83.3	12	00001011	8	1	3	2	2	0100	001	75.0
83.3	12	00001011	8	1	5	1	1	0101	000	87.5
33.3	10	00001001	24	1	7	8	8	1110	111	66.7
33.3	10	00001001	24	1	9	7	7	1111	110	70.8
33.3	12	00001011	20	1	7	6	6	1100	101	70.0
33.3	12	00001011	20	1	9	5	5	1101	100	75.0
33.3	15	00001110	16	1	7	4	4	1010	011	75.0
33.3	15	00001110	16	1	9	3	3	1011	010	81.3
33.3	16	00001111	15	1	6	4	4	1001	011	73.3
33.3	16	00001111	15	1	8	3	3	1010	010	80.0
33.3	20	00010011	12	1	5	3	3	0111	010	75.0
33.3	20	00010011	12	1	7	2	2	1000	001	83.3
33.3	24	00010111	10	1	3	3	3	0101	010	70.0
33.3	24	00010111	10	1	5	2	2	0110	001	80.0
33.3	30	00011101	8	1	3	2	2	0100	001	75.0
33.3	30	00011101	8	1	5	1	1	0101	000	87.5

Caution The values in Table 13-23 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Table 13-24. Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (1/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of CBRP	CBRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					CBTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	16	1	1	7	7	0111	110	56.3
1000	1	00000000	16	1	3	6	6	1000	101	62.5
1000	1	00000000	16	1	5	5	5	1001	100	68.8
1000	1	00000000	16	1	7	4	4	1010	011	75.0
1000	1	00000000	16	1	9	3	3	1011	010	81.3
1000	1	00000000	16	1	11	2	2	1100	001	87.5
1000	1	00000000	16	1	13	1	1	1101	000	93.8
1000	2	00000001	8	1	3	2	2	0100	001	75.0
1000	2	00000001	8	1	5	1	1	0101	000	87.5
500	2	00000001	16	1	1	7	7	0111	110	56.3
500	2	00000001	16	1	3	6	6	1000	101	62.5
500	2	00000001	16	1	5	5	5	1001	100	68.8
500	2	00000001	16	1	7	4	4	1010	011	75.0
500	2	00000001	16	1	9	3	3	1011	010	81.3
500	2	00000001	16	1	11	2	2	1100	001	87.5
500	2	00000001	16	1	13	1	1	1101	000	93.8
500	4	00000011	8	1	3	2	2	0100	001	75.0
500	4	00000011	8	1	5	1	1	0101	000	87.5
250	4	00000011	16	1	3	6	6	1000	101	62.5
250	4	00000011	16	1	5	5	5	1001	100	68.8
250	4	00000011	16	1	7	4	4	1010	011	75.0
250	4	00000011	16	1	9	3	3	1011	010	81.3
250	4	00000011	16	1	11	2	2	1100	001	87.5
250	8	00000111	8	1	3	2	2	0100	001	75.0
250	8	00000111	8	1	5	1	1	0101	000	87.5
125	8	00000111	16	1	3	6	6	1000	101	62.5
125	8	00000111	16	1	7	4	4	1010	011	75.0
125	8	00000111	16	1	9	3	3	1011	010	81.3
125	8	00000111	16	1	11	2	2	1100	001	87.5
125	16	00001111	8	1	3	2	2	0100	001	75.0
125	16	00001111	8	1	5	1	1	0101	000	87.5

Caution The values in Table 13-24 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Table 13-24. Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (2/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of CBRP	CBRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					CBTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
100	8	00000111	20	1	9	5	5	1101	100	75.0
100	8	00000111	20	1	11	4	4	1110	011	80.0
100	10	00001001	16	1	7	4	4	1010	011	75.0
100	10	00001001	16	1	9	3	3	1011	010	81.3
100	16	00001111	10	1	3	3	3	0101	010	70.0
100	16	00001111	10	1	5	2	2	0110	001	80.0
100	20	00010011	8	1	3	2	2	0100	001	75.0
83.3	8	00000111	24	1	7	8	8	1110	111	66.7
83.3	8	00000111	24	1	9	7	7	1111	110	70.8
83.3	12	00001011	16	1	7	4	4	1010	011	75.0
83.3	12	00001011	16	1	9	3	3	1011	010	81.3
83.3	12	00001011	16	1	11	2	2	1100	001	87.5
83.3	16	00001111	12	1	5	3	3	0111	010	75.0
83.3	16	00001111	12	1	7	2	2	1000	001	83.3
83.3	24	00010111	8	1	3	2	2	0100	001	75.0
83.3	24	00010111	8	1	5	1	1	0101	000	87.5
33.3	30	00011101	24	1	7	8	8	1110	111	66.7
33.3	30	00011101	24	1	9	7	7	1111	110	70.8
33.3	24	00010111	20	1	9	5	5	1101	100	75.0
33.3	24	00010111	20	1	11	4	4	1110	011	80.0
33.3	30	00011101	16	1	7	4	4	1010	011	75.0
33.3	30	00011101	16	1	9	3	3	1011	010	81.3
33.3	32	00011111	15	1	8	3	3	1010	010	80.0
33.3	32	00011111	15	1	10	2	2	1011	001	86.7
33.3	37	00100100	13	1	6	3	3	1000	010	76.9
33.3	37	00100100	13	1	8	2	2	1001	001	84.6
33.3	40	00100111	12	1	5	3	3	0111	010	75.0
33.3	40	00100111	12	1	7	2	2	1000	001	83.3
33.3	48	00101111	10	1	3	3	3	0101	010	70.0
33.3	48	00101111	10	1	5	2	2	0110	001	80.0
33.3	60	00111011	8	1	3	2	2	0100	001	75.0
33.3	60	00111011	8	1	5	1	1	0101	000	87.5

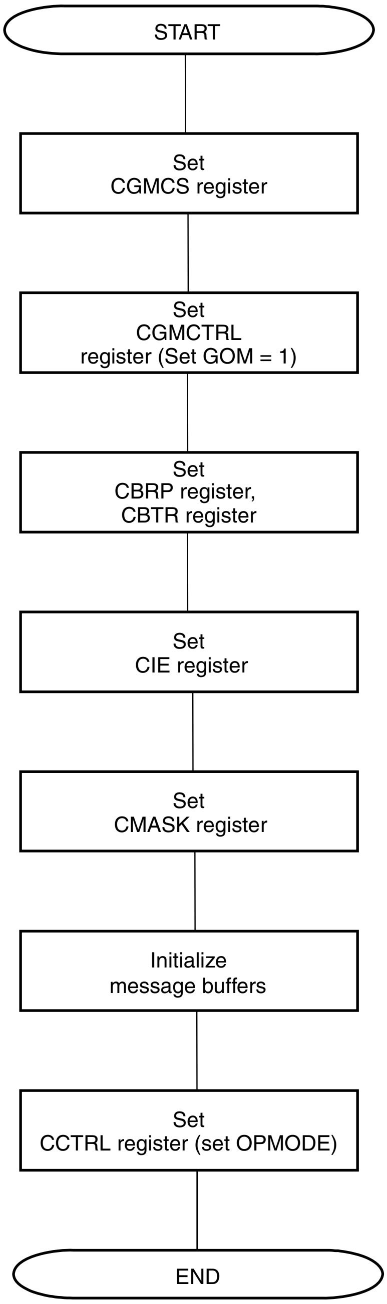
Caution The values in Table 13-24 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

13.16 Operation of CAN Controller

The processing procedure for showing in this chapter is recommended processing procedure to operate CAN controller. Develop the program referring to recommended processing procedure in this chapter.

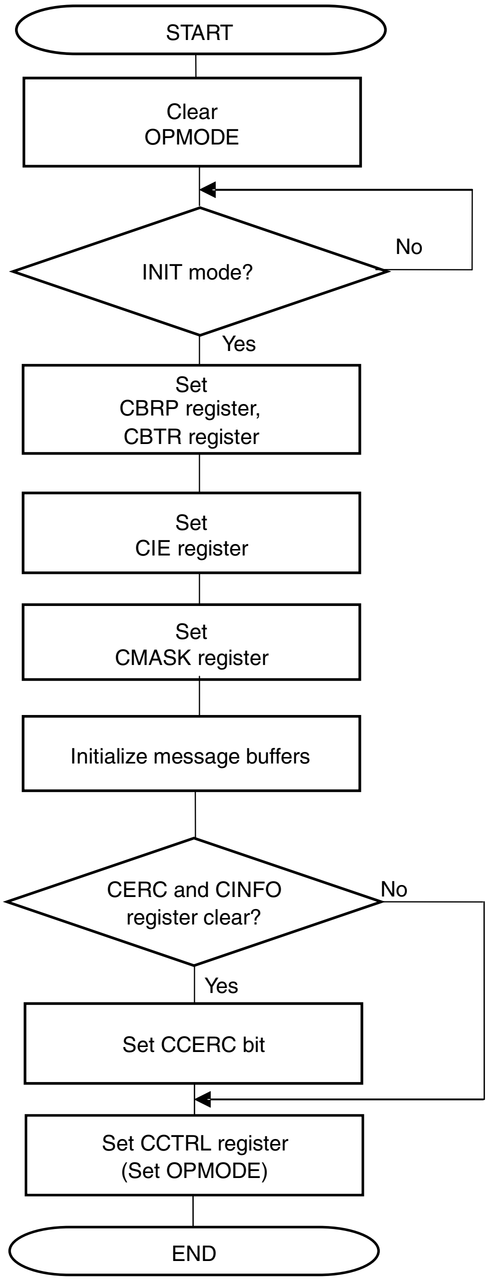
Remark m = 0 to 15

Figure 13-63. Initialization



Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

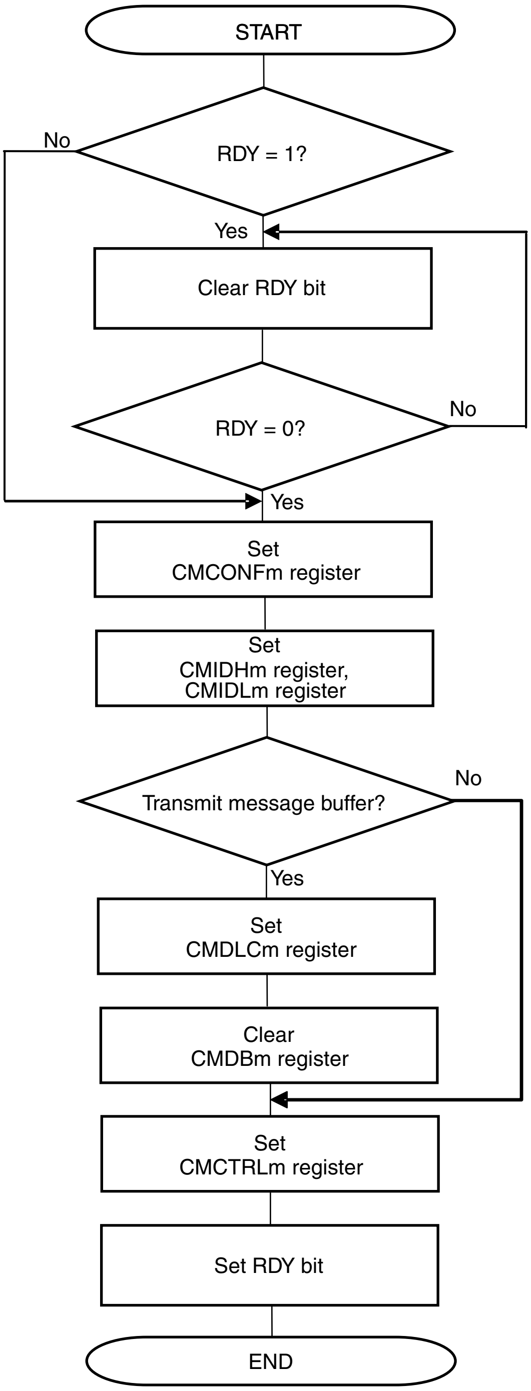
Figure 13-64. Re-initialization



Caution After setting the CAN module to the initialization mode, avoid setting the module to another operation mode immediately after. If it is necessary to immediately set the module to another operation mode, be sure to access registers other than the CCTRL and CGMCTRL registers (e.g. set a message buffer).

Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 13-65. Message Buffer Initialization



Cautions 1. Before a message buffer is initialized, the RDY bit must be cleared.

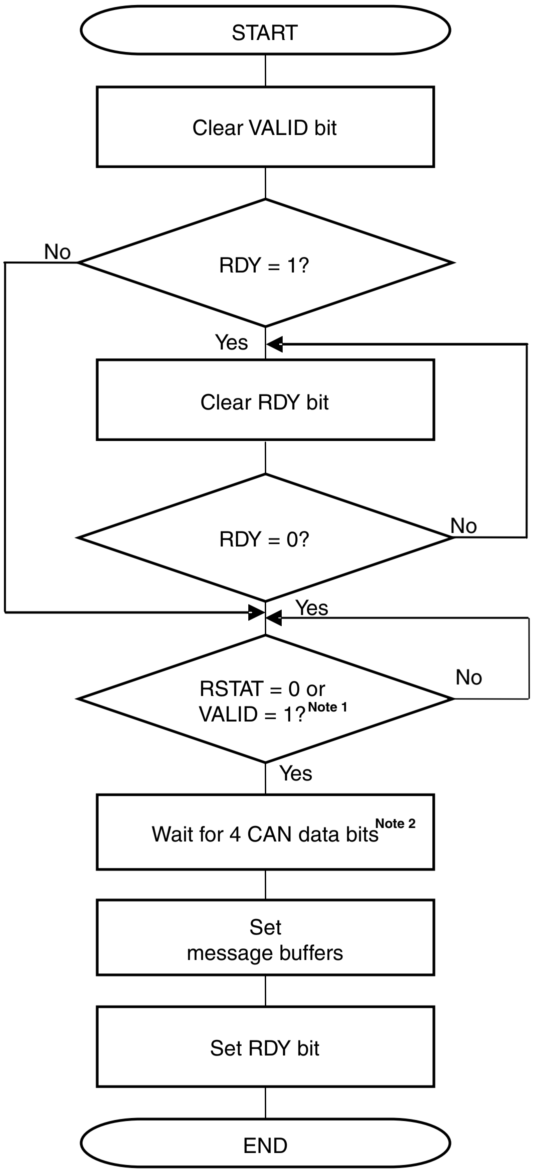
2. Make the following settings for message buffers not used by the application.

- Clear the RDY, TRQ, and DN bits of the CMCTRLm register to 0.

- Clear the MA0 bit of the CMCONFm register to 0.

Figure 13-66 shows the processing for a receive message buffer (MT[2:0] bits of CMCONFm register = 001B to 101B).

Figure 13-66. Message Buffer Redefinition



Notes 1. Confirm that a message is being received because RDY bit must be set after a message is completely received.

2. Avoid message buffer redefinition during store operation of message reception by waiting additional 4 CAN data bits.

Figure 13-67 shows the processing for a transmit message buffer during transmission (MT[2:0] bits of CMCONFm register = 000B).

Figure 13-67. Message Buffer Redefinition during Transmission

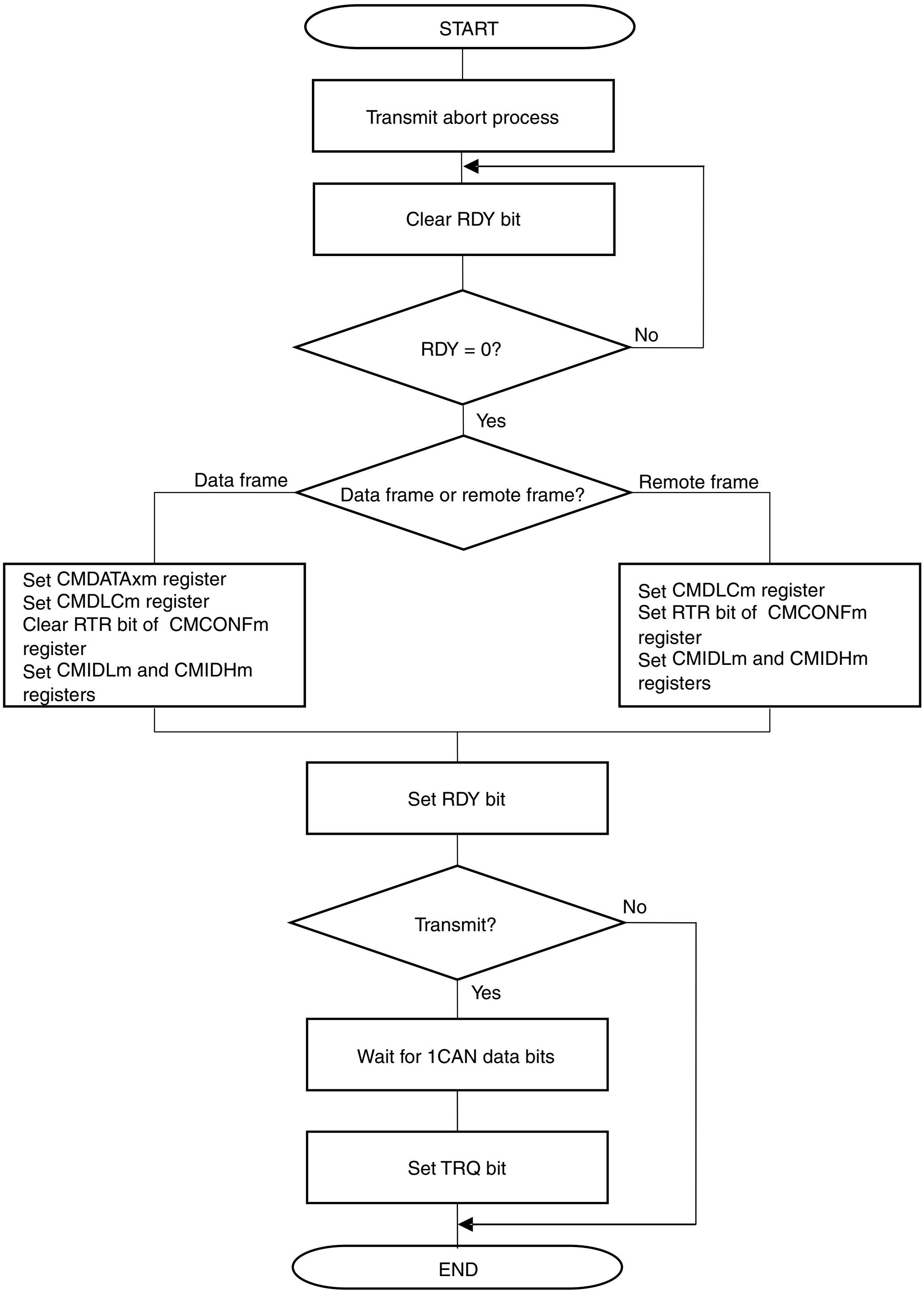
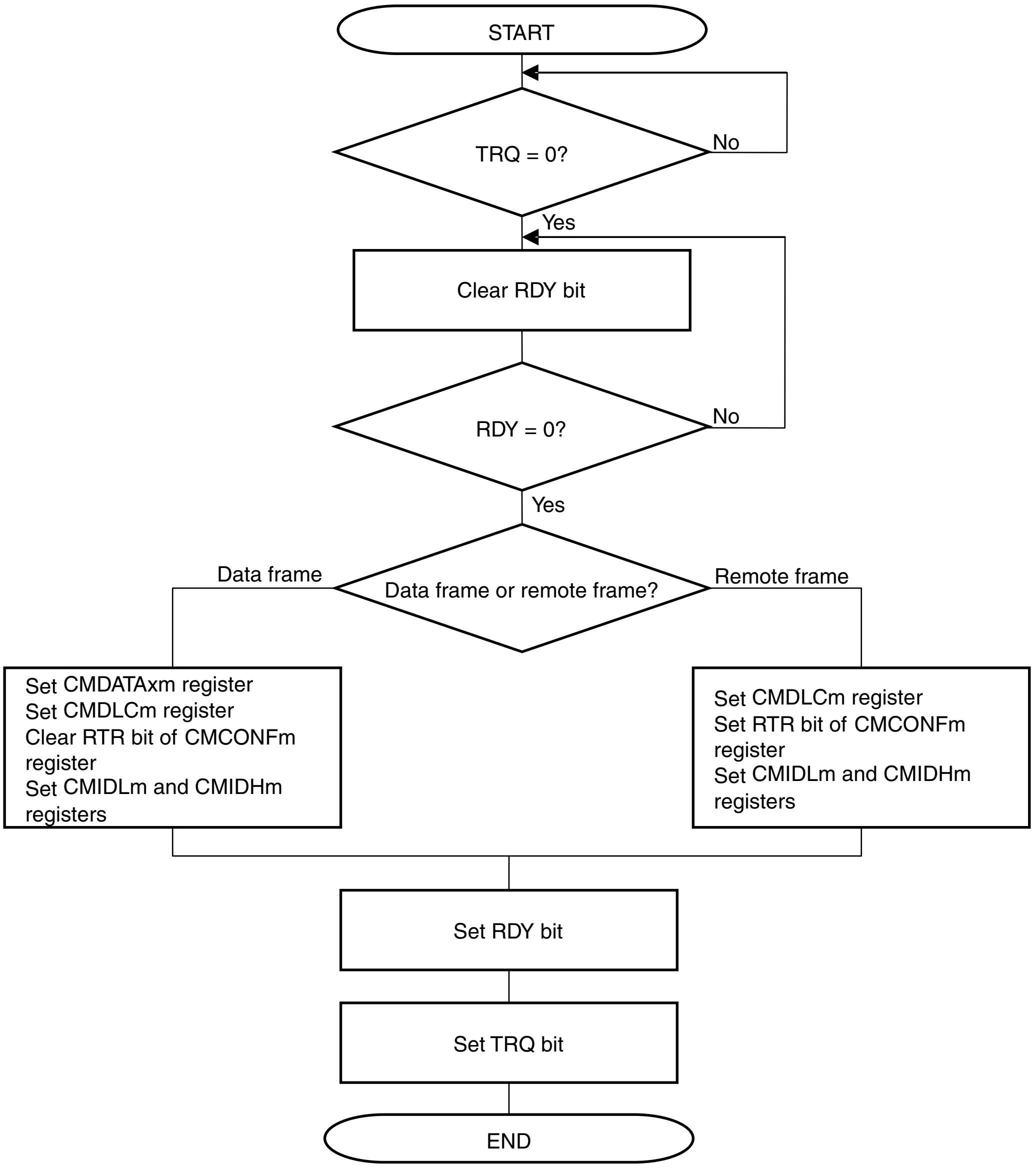


Figure 13-68 shows the processing for a transmit message buffer (MT[2:0] bits of CMCONFm register = 000B).

Figure 13-68. Message Transmit Processing

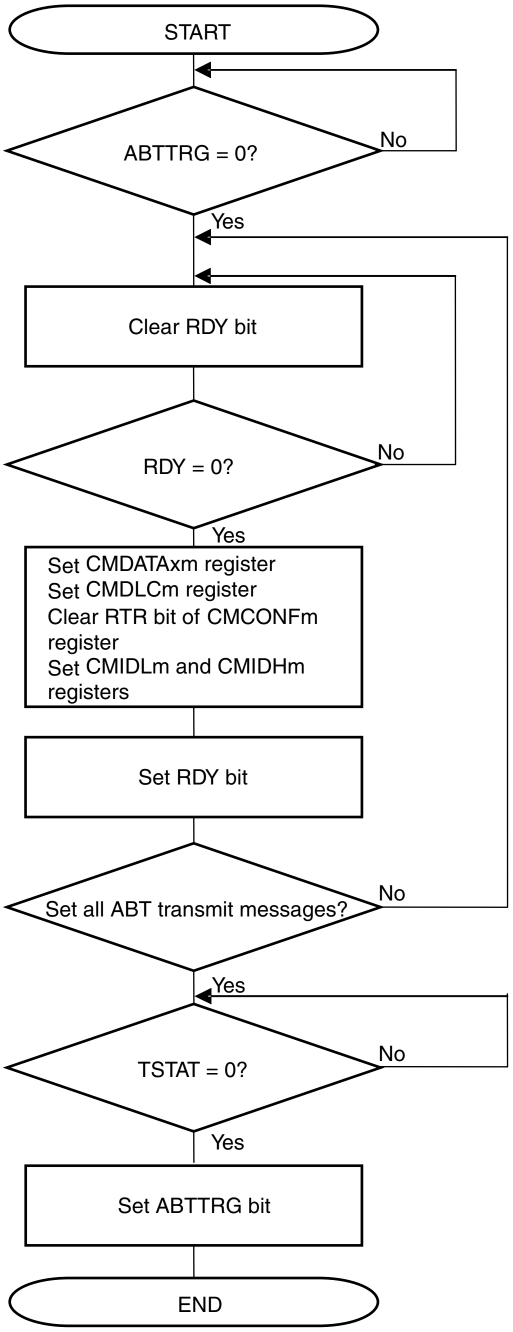

START
TRQ = 0?
No
Yes
Clear RDY bit
RDY = 0?
No
Yes
Data frame
Data frame or remote frame?
Remote frame
Set CMDATAxm register
Set CMDLCm register
Clear RTR bit of CMCONFm register
Set CMIDLm and CMIDHm registers
Set CMDLCm register
Set RTR bit of CMCONFm register
Set CMIDLm and CMIDHm registers
Set RDY bit
Set TRQ bit
END

Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Figure 13-69 shows the processing for a transmit message buffer (MT[2:0] bits of CMCONFm register = 000B).

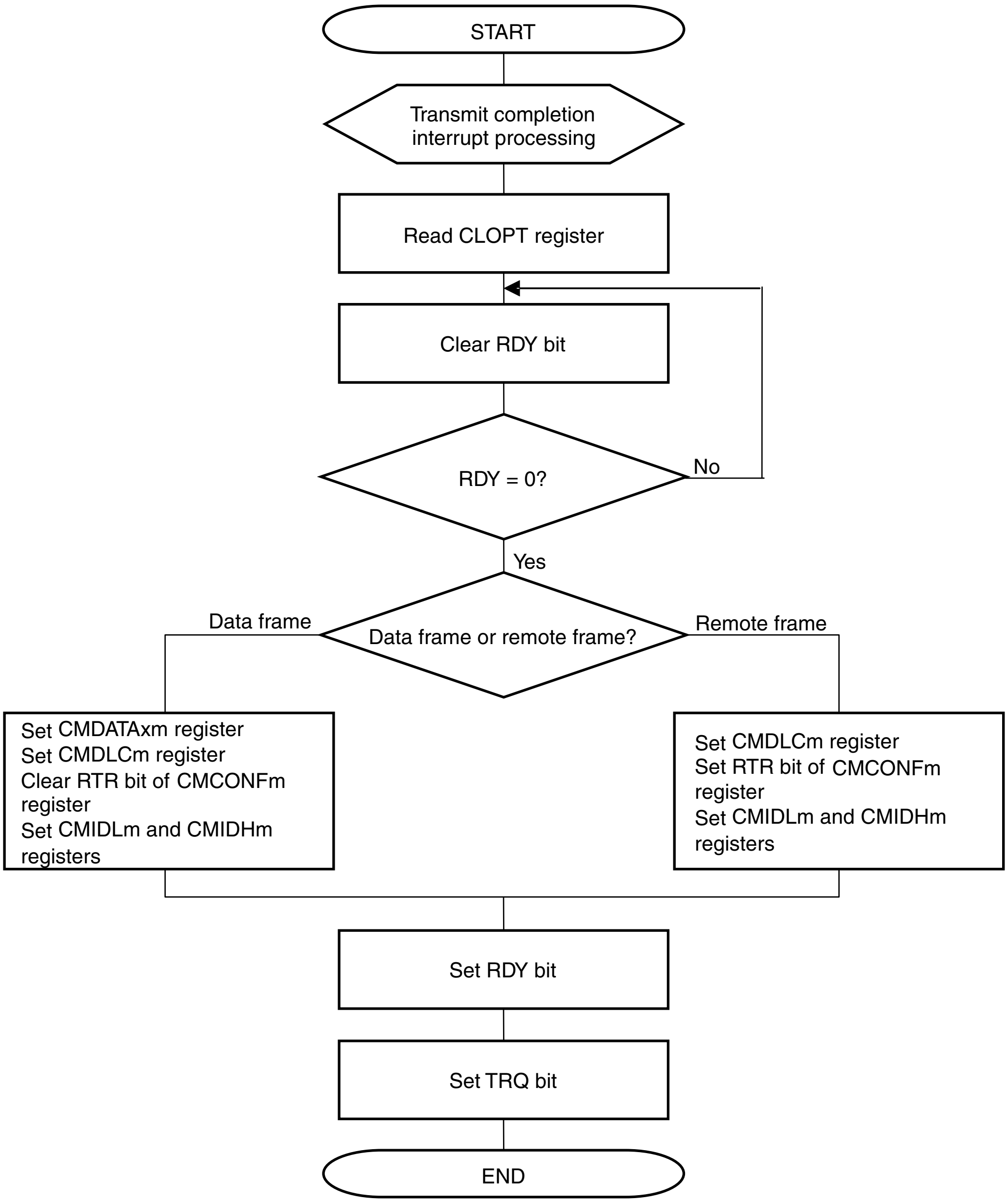
Figure 13-69. ABT Message Transmit Processing



Caution The ABTTRG bit should be set to 1 after the TSTAT bit is cleared to 0. Checking the TSTAT bit and setting the ABTTRG bit to 1 must be processed continuously.

Remark This processing (normal operation mode with ABS) can only be applied to message buffers 0 to 7. For message buffers other than the ABT message buffers, see **Figure 13-68**.

Figure 13-70. Transmission via Interrupt (Using CLOPT Register)


START
Transmit completion interrupt processing
Read CLOPT register
Clear RDY bit
RDY = 0?
No
Yes
Data frame or remote frame?
Data frame
Remote frame
Set CMDATAxm register
Set CMDLCm register
Clear RTR bit of CMCONFm register
Set CMIDLm and CMIDHm registers
Set CMDLCm register
Set RTR bit of CMCONFm register
Set CMIDLm and CMIDHm registers
Set RDY bit
Set TRQ bit
END

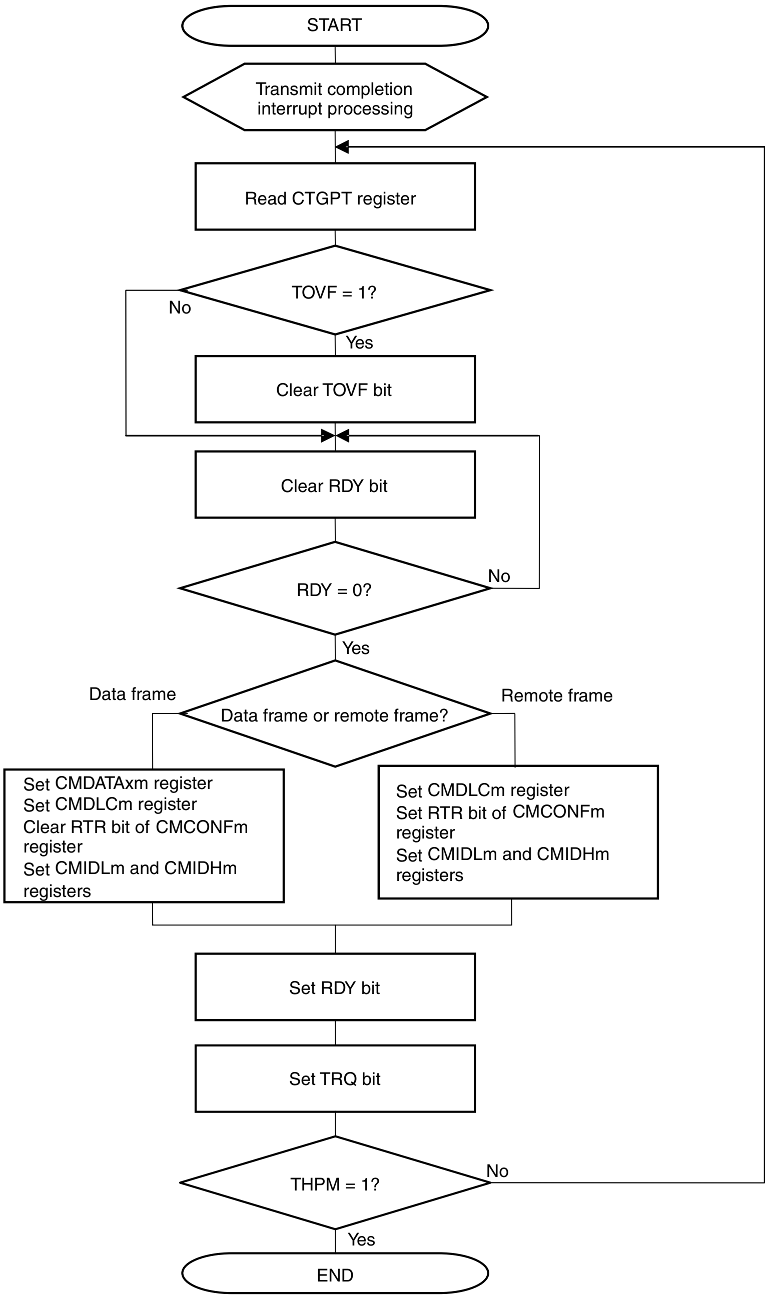
Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Remark Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.

It is recommended to cancel any sleep mode requests, before processing TX interrupts.

Figure 13-71. Transmit via Interrupt (Using CTGPT Register)



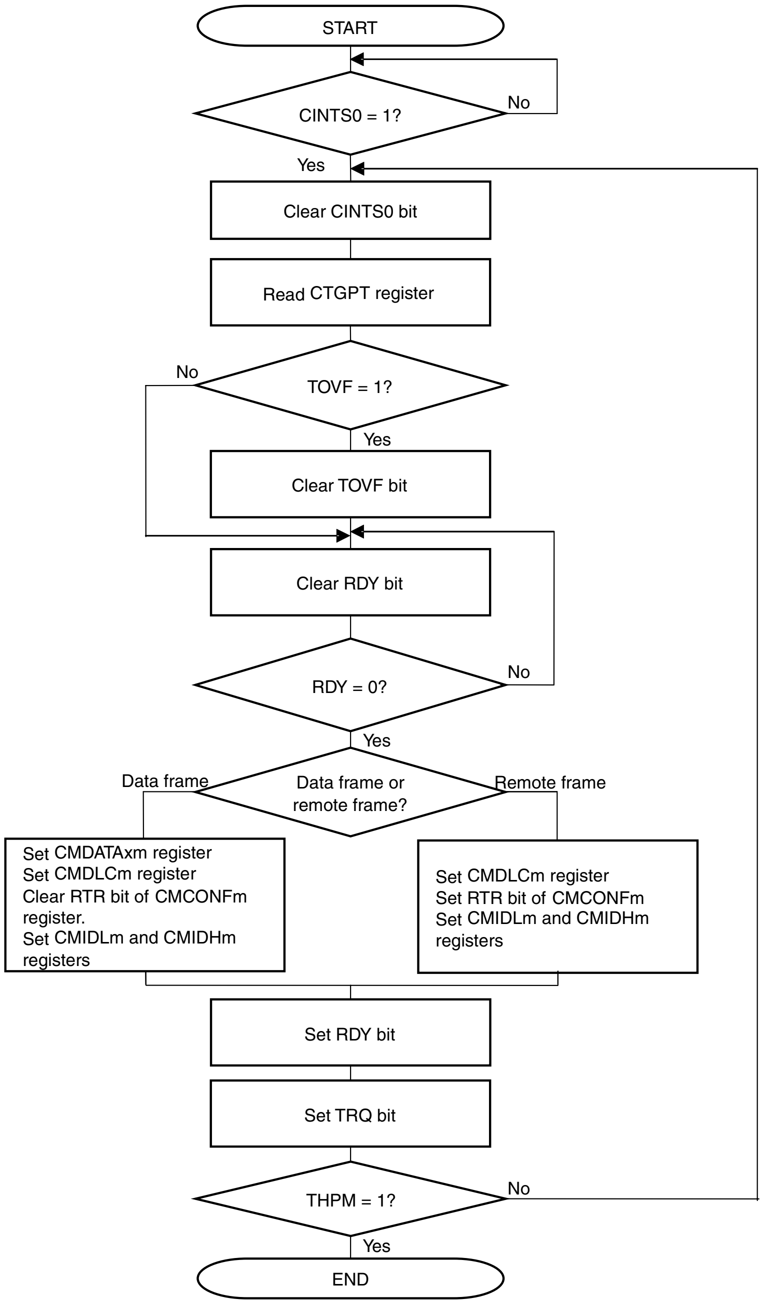
Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Remarks 1. Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
It is recommended to cancel any sleep mode requests, before processing TX interrupts.

2. If TOVF was set once, the transmit history list is inconsistent. Consider to scan all configured transmit buffers for completed transmissions.

Figure 13-72. Transmission via Software Polling



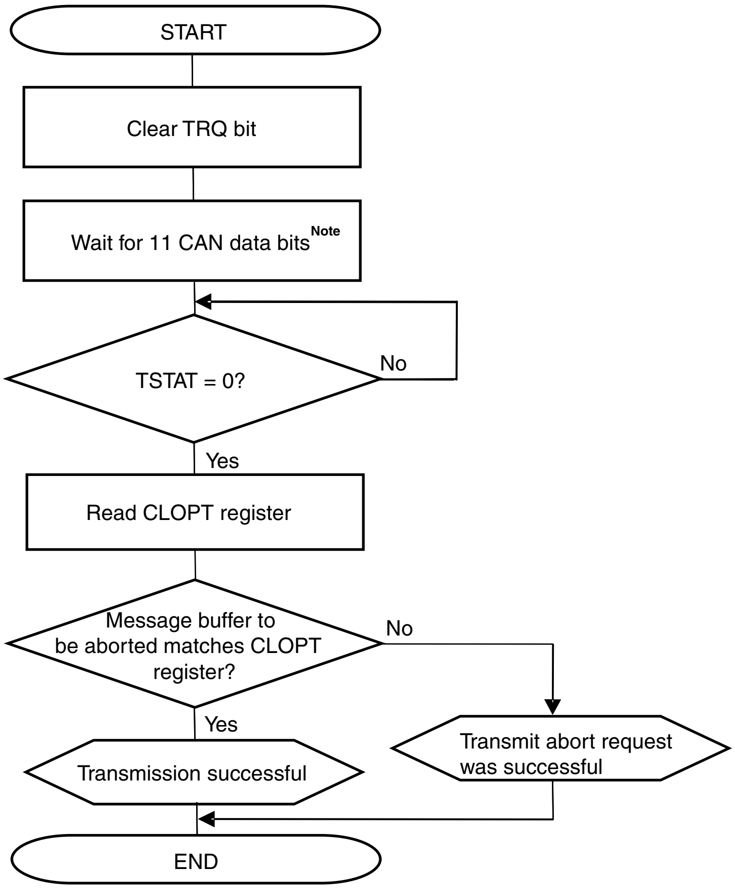
Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Remarks 1. Also check the MBON flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.

2. If TOVF was set once, the transmit history list is inconsistent. Consider to scan all configured transmit buffers for completed transmissions.

Figure 13-73. Transmission Abort Processing (Except Normal Operation Mode with ABT)


START
Clear TRQ bit
Wait for 11 CAN data bits Note
TSTAT = 0?
No
Yes
Read CLOPT register
Message buffer to be aborted matches CLOPT register?
No
Yes
Transmission successful
Transmit abort request was successful
END

Note There is a possibility of starting the transmission without being aborted even if TRQ bit is cleared, because the transmission request to protocol layer might already been accepted between 11 bits, total of interframe space (3 bits) and suspend transmission (8 bits).

Cautions 1. Execute transmission request abort processing by clearing the TRQ bit, not the RDY bit.

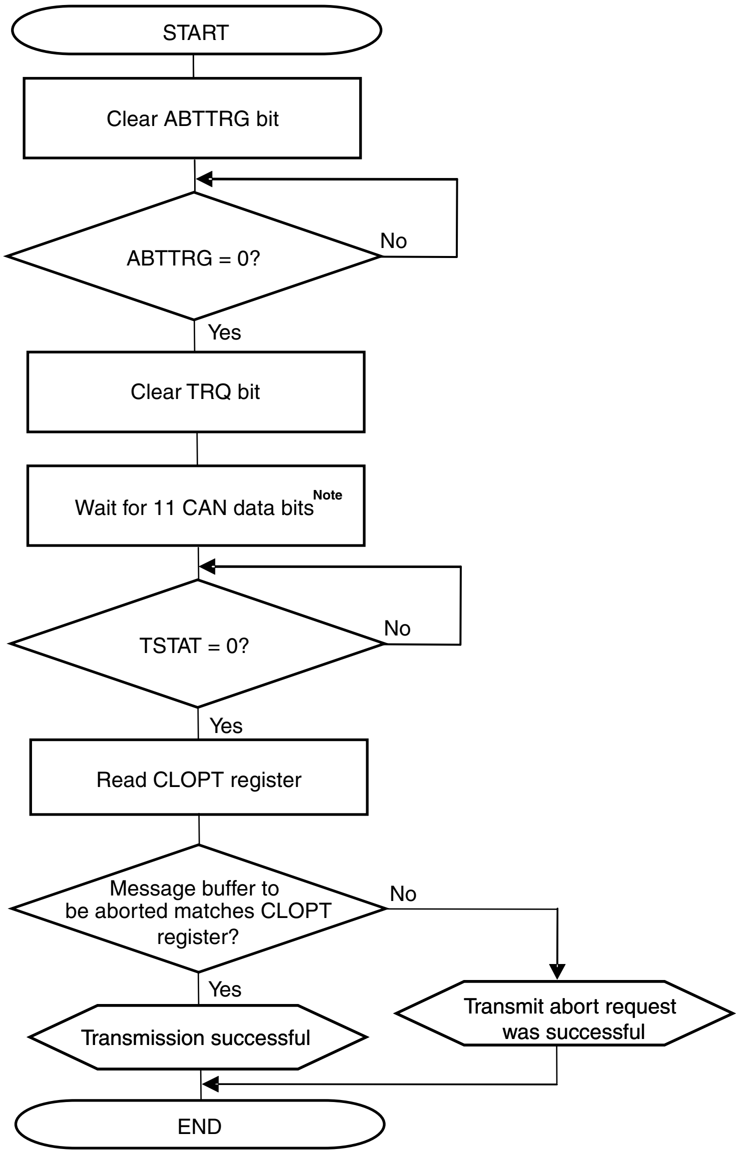
2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.

3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute the new transmission request including in the other message buffers while transmission abort processing is in progress.

5. There is a possibility that contradiction is caused in the judgment whether the transmission abort request was successful when the transmission from the same message buffer is consecutive or only one message buffer is used. In that case, judge it by using the history information etc. that the CTGPT register indicates.

Figure 13-74. Transmission Abort Processing Except for ABT Transmission (Normal Operation Mode with ABT)


START
Clear ABTTRG bit
ABTTRG = 0?
No
Yes
Clear TRQ bit
Wait for 11 CAN data bits Note
TSTAT = 0?
No
Yes
Read CLOPT register
Message buffer to be aborted matches CLOPT register?
No
Yes
Transmission successful
Transmit abort request was successful
END

Note There is a possibility of starting the transmission without being aborted even if TRQ bit is cleared, because the transmission request to protocol layer might already been accepted between 11 bits, total of interframe space (3 bits) and suspend transmission (8 bits).

Cautions 1. Execute transmission request abort processing by clearing the TRQ bit, not the RDY bit.

2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.

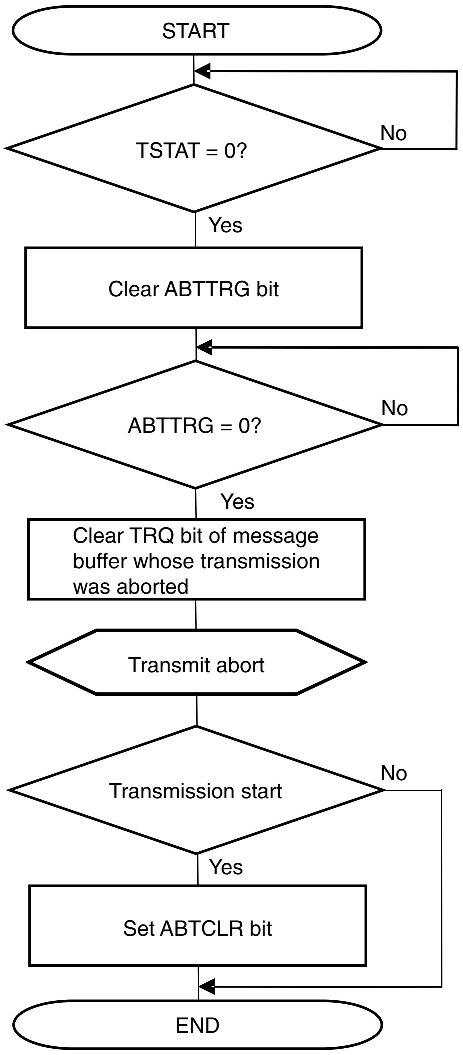
3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute the new transmission request including in the other message buffers while transmission abort processing is in progress.

5. There is a possibility that contradiction is caused in the judgment whether the transmission abort request was successful when the transmission from the same message buffer is consecutive or only one message buffer is used. In that case, judge it by using the history information etc. that the CnTGPT register indicates.

Figure 13-75 shows the processing not to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

Figure 13-75. ABT Transmission Abort Processing (Normal Operation Mode with ABT)

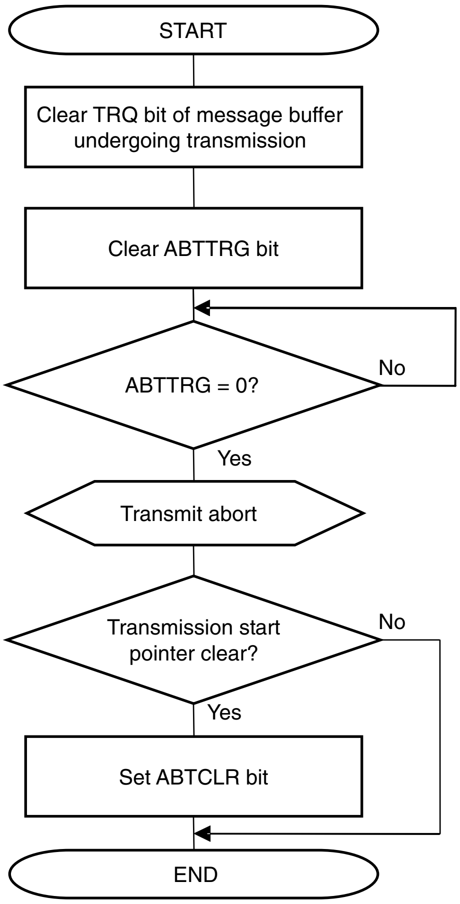


Cautions 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. Make a CAN sleep mode/CAN stop mode transition request after ABTTRG bit is cleared (after ABT mode is aborted) following the procedure shown in Figure 13-75 or 13-76. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 13-74.

Figure 13-76 shows the processing to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

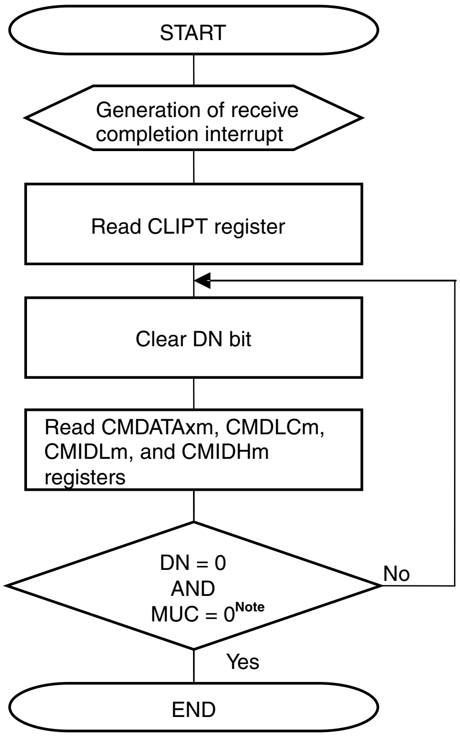
Figure 13-76. ABT Transmission Request Abort Processing (Normal Operation Mode with ABT)


START
Clear TRQ bit of message buffer undergoing transmission
Clear ABTTRG bit
ABTTRG = 0?
No
Yes
Transmit abort
Transmission start pointer clear?
No
Yes
Set ABTCLR bit
END

Cautions 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. Make a CAN sleep mode/CAN stop mode request after ABTTRG is cleared (after ABT mode is stopped) following the procedure shown in Figure 13-75 or 13-76. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 13-74.

Figure 13-77. Reception via Interrupt (Using CLIPT Register)

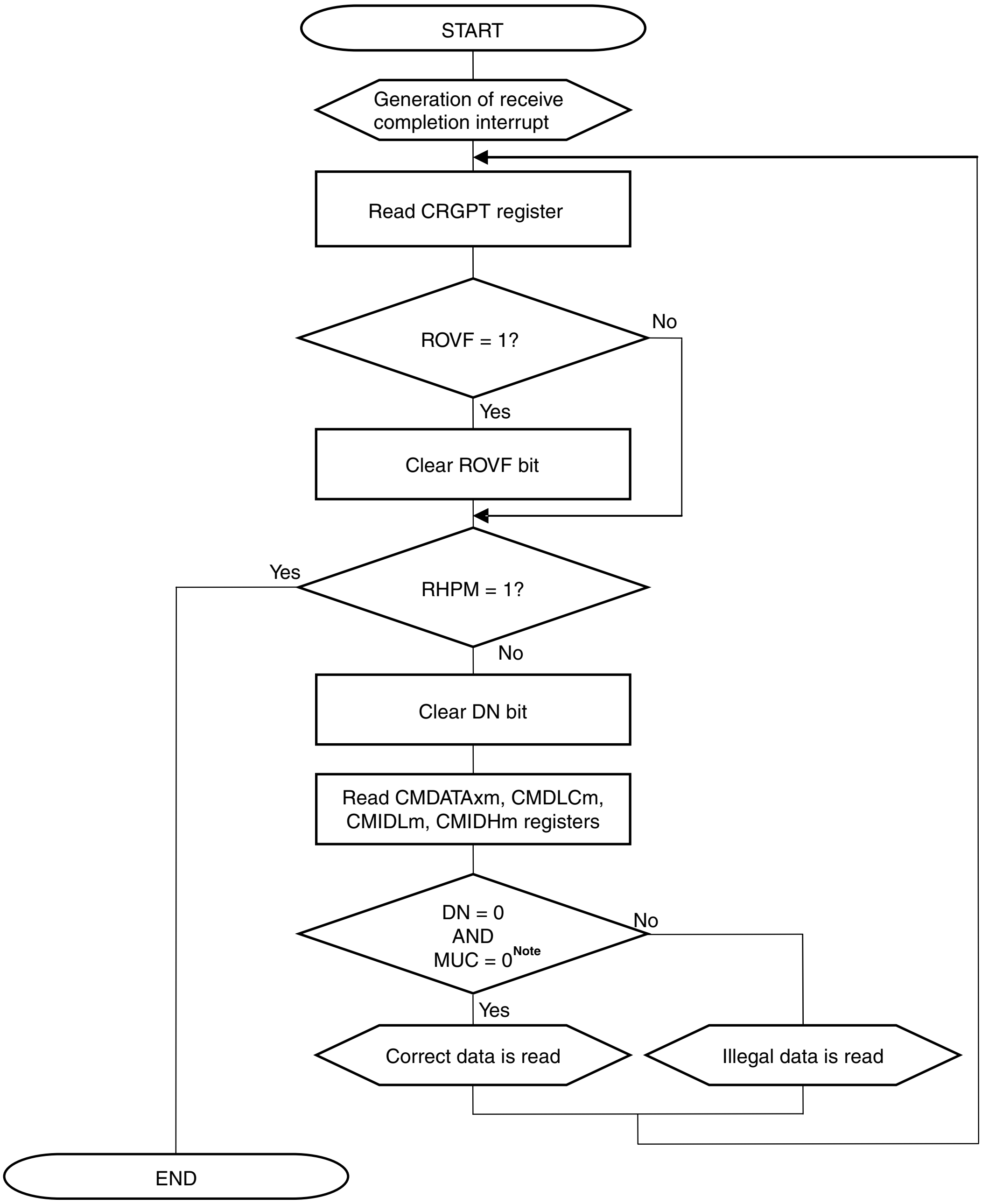

START
Generation of receive completion interrupt
Read CLIPT register
Clear DN bit
Read CMDATAxm, CMDLCm, CMIDLm, and CMIDHm registers
DN = 0
AND
MUC = 0Note
No
Yes
END

Note Check the MUC and DN bits using one read access.

Remark Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.

It is recommended to cancel any sleep mode requests, before processing RX interrupts.

Figure 13-78. Reception via Interrupt (Using CRGPT Register)

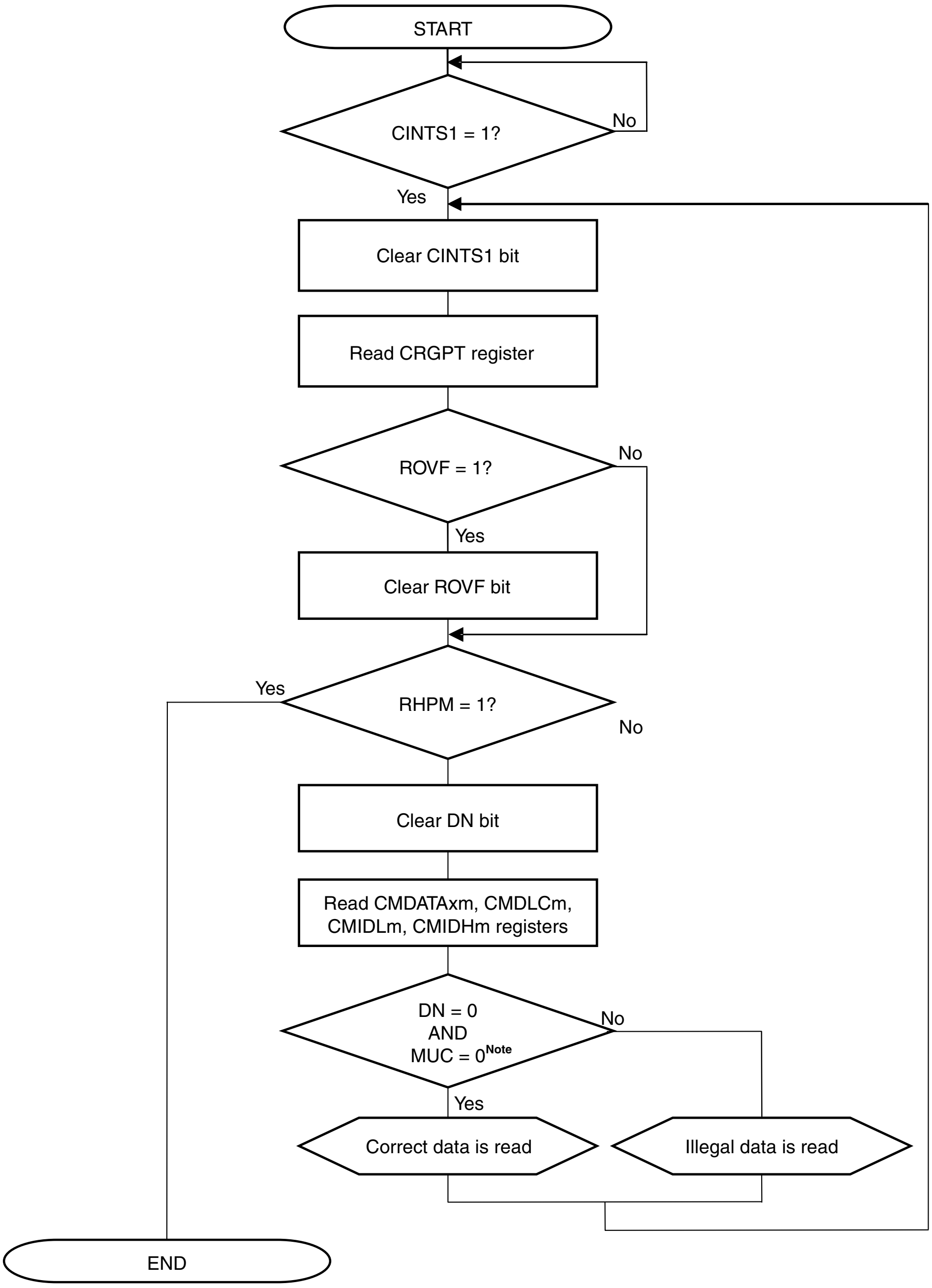


Note Check the MUC and DN bits using one read access.

Remarks 1. Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
It is recommended to cancel any sleep mode requests, before processing RX interrupts.

2. If ROVF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.

Figure 13-79. Reception via Software Polling

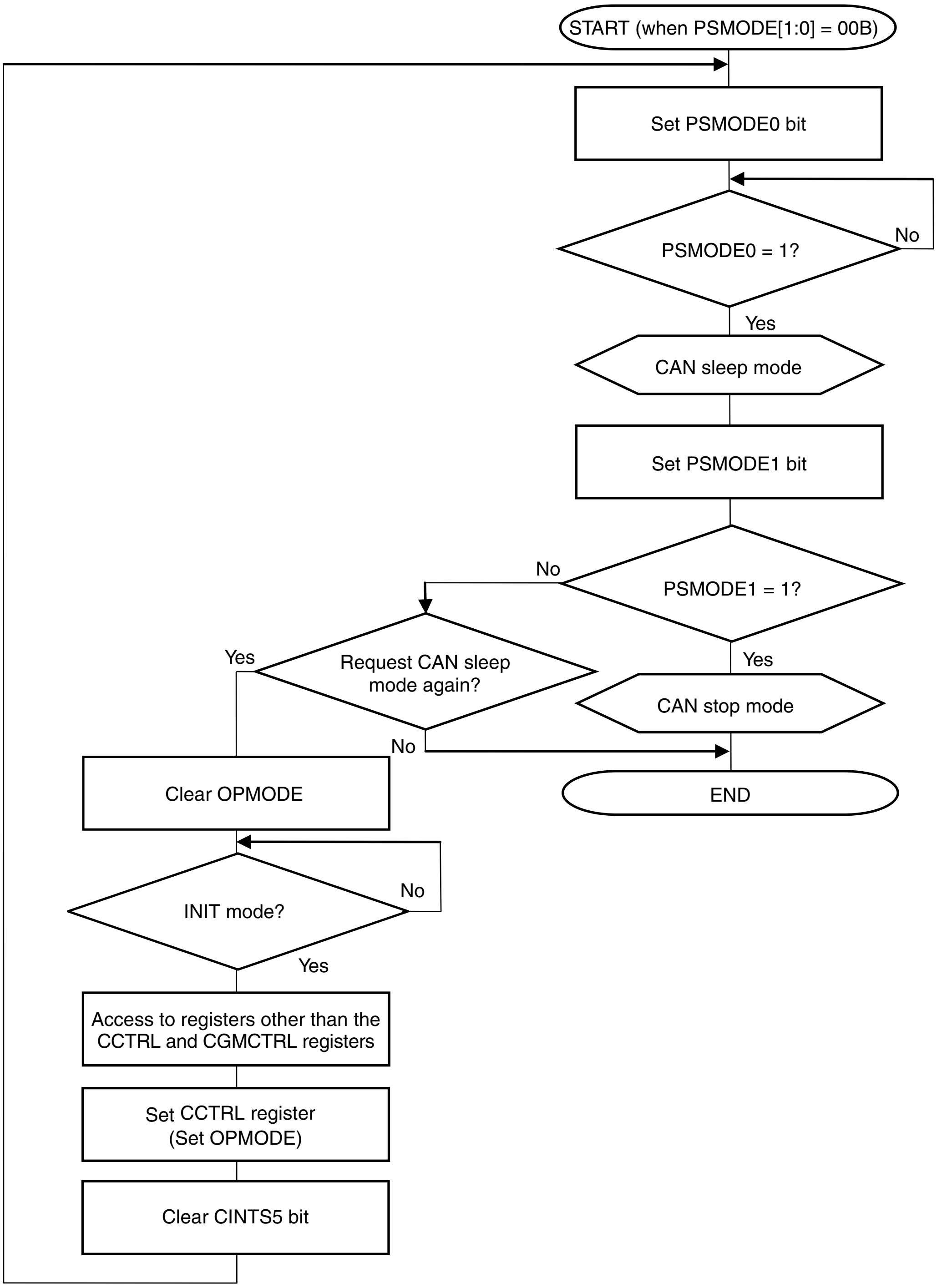


Note Check the MUC and DN bits using one read access.

Remarks 1. Also check the MBON flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.

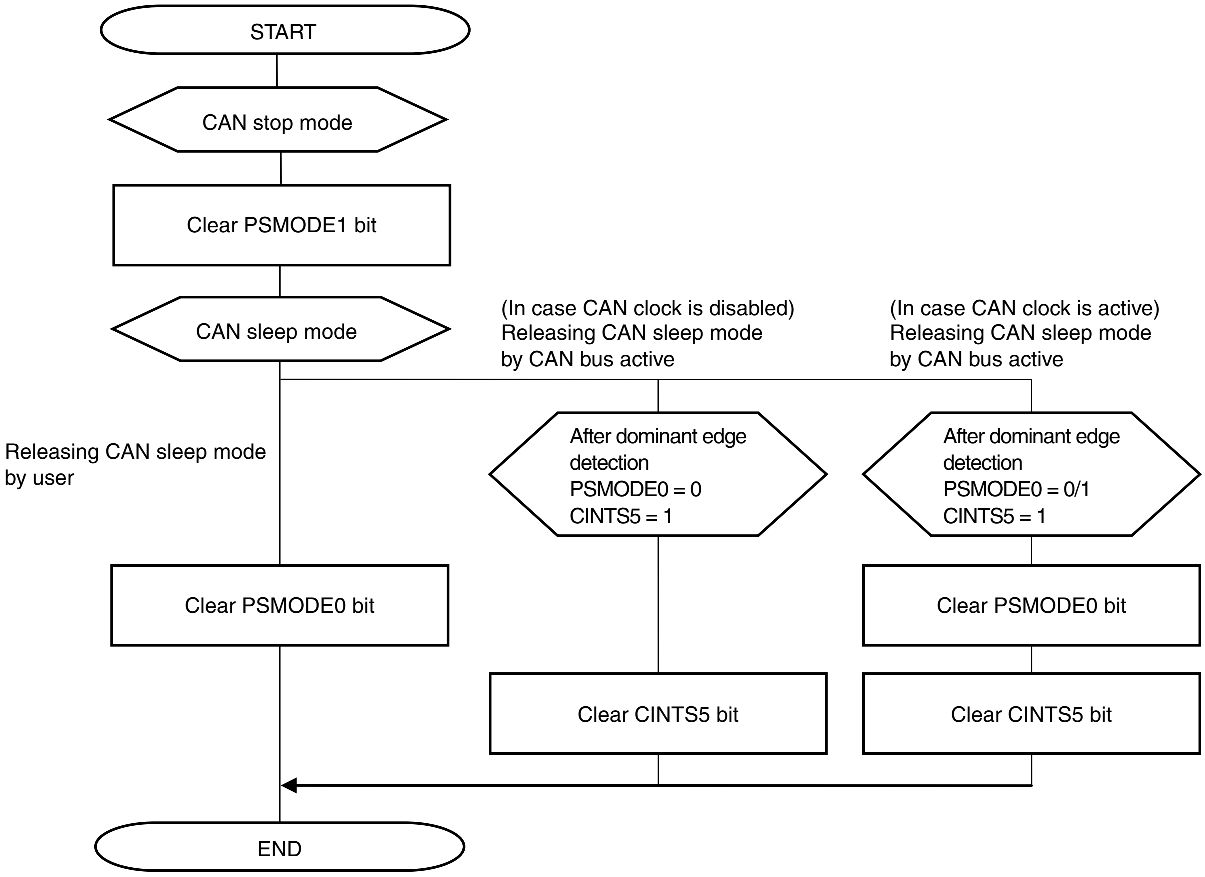
2. If ROVF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.

Figure 13-80. Setting CAN Sleep Mode/Stop Mode


START (when PSMODE[1:0] = 00B)
Set PSMODE0 bit
PSMODE0 = 1?
No
Yes
CAN sleep mode
Set PSMODE1 bit
PSMODE1 = 1?
No
Yes
CAN stop mode
Request CAN sleep mode again?
Yes
No
END
Clear OPMODE
INIT mode?
No
Yes
Access to registers other than the CCTRL and CGMCTRL registers
Set CCTRL register (Set OPMODE)
Clear CINTS5 bit

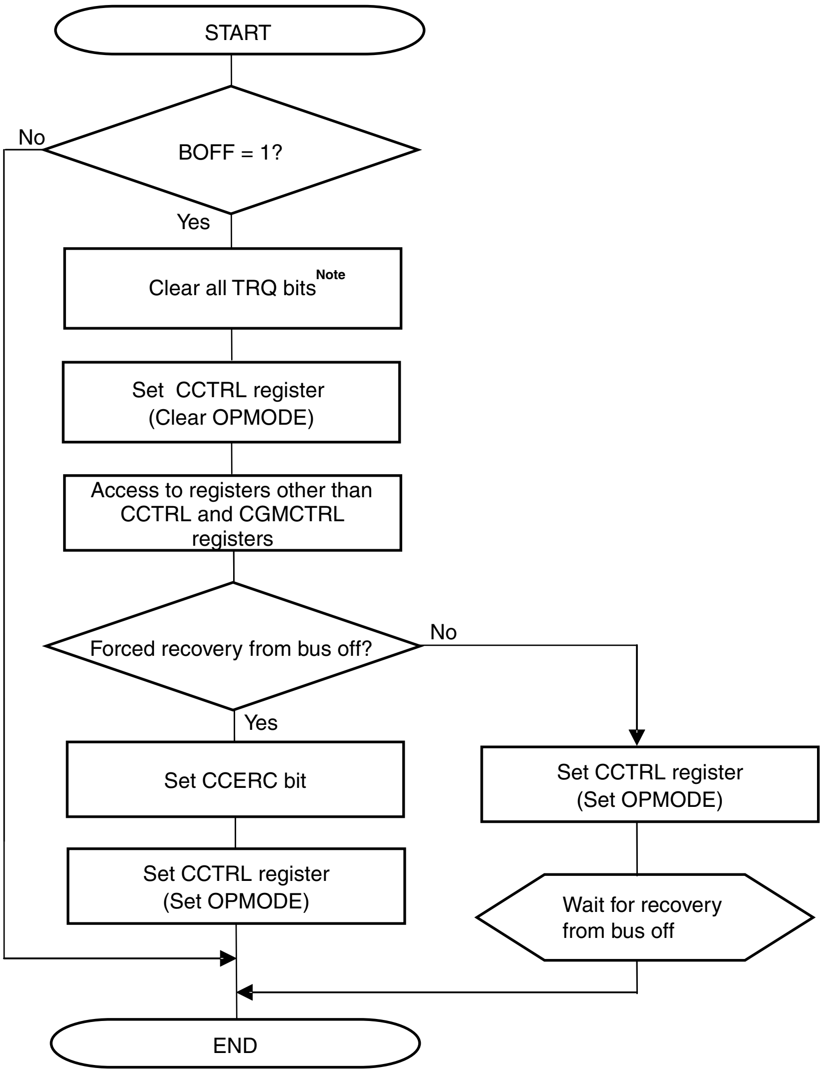
Caution To abort transmission before making a request for the CAN sleep mode, perform processing according to Figure 13-73 or 13-74.

Figure 13-81. Clear CAN Sleep/Stop Mode



Remark “In case CAN clock is disabled”: By means of the CPU standby mode, the CAN module clock has been switched off, and the CAN module is in sleep mode.

Figure 13-82. Bus-Off Recovery (Expect Normal Operation Mode with ABT)

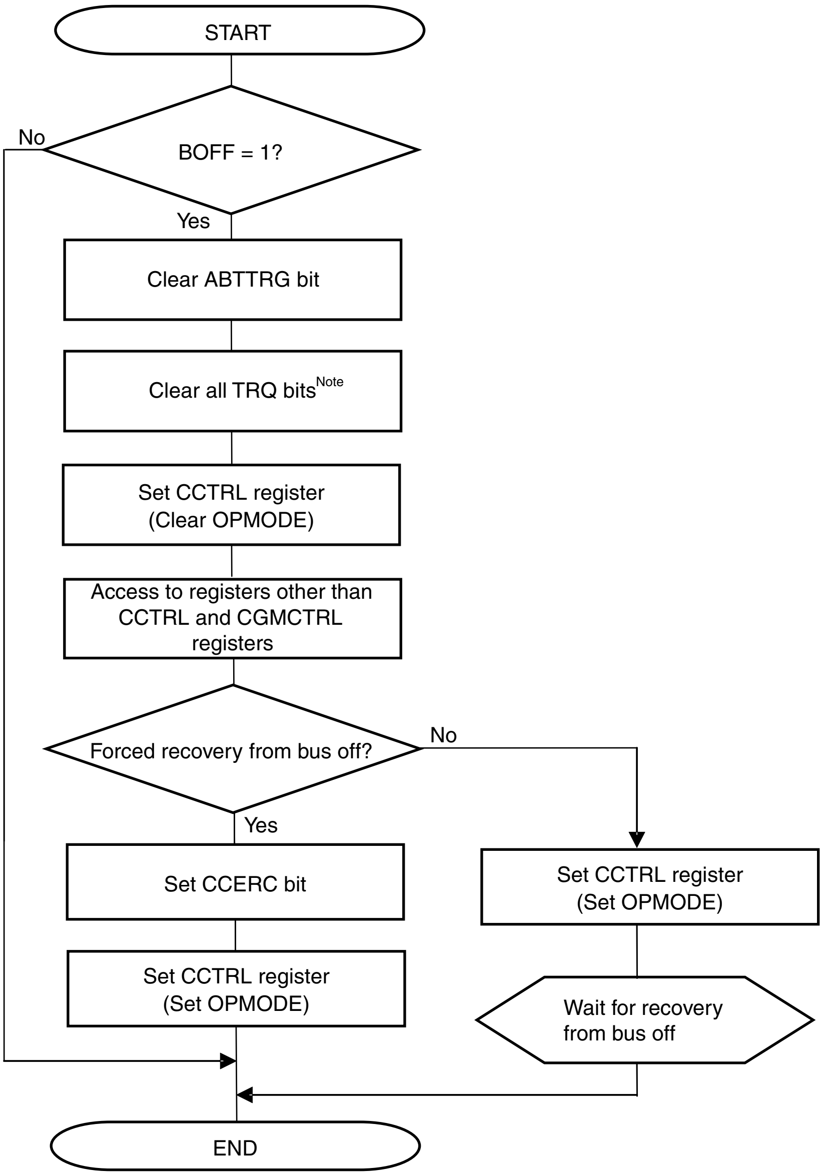


Note Clear all TRQ bits when re-initialization of message buffer is executed by clearing RDY bit before bus-off recovery sequence is started.

Caution When the transmission from the initialization mode to any operation modes is requested to execute bus-off recovery sequence again in the bus-off recovery sequence, reception error counter is cleared. Therefore it is necessary to detect 11 consecutive recessive-level bits 128 times on the bus again.

Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 13-83. Bus-Off Recovery (Normal Operation Mode with ABT)


START
BOFF = 1?
No
Yes
Clear ABTTRG bit
Clear all TRQ bits[Note]
Set CCTRL register (Clear OPMODE)
Access to registers other than CCTRL and CGMCTRL registers
Forced recovery from bus off?
No
Yes
Set CCERC bit
Set CCTRL register (Set OPMODE)
Set CCTRL register (Set OPMODE)
Wait for recovery from bus off
END

Note Clear all TRQ bits when re-initialization of message buffer is executed by clearing RDY bit before bus-off recovery sequence is started.

Caution When the transmission from the initialization mode to any operation modes is requested to execute bus-off recovery sequence again in the bus-off recovery sequence, reception error counter is cleared. Therefore it is necessary to detect 11 consecutive recessive-level bits 128 times on the bus again.

Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 13-84. Normal Shutdown Process

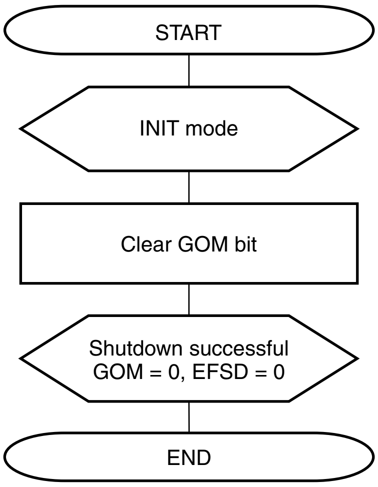
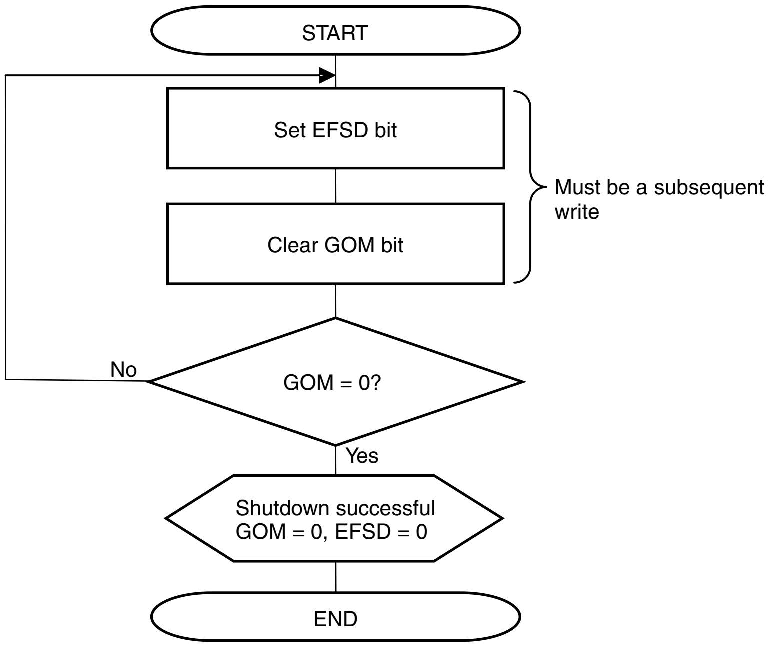


Figure 13-85. Forced Shutdown Process



<R> **Caution Do not read- or write-access any registers by software between setting the EFSD bit and clearing the GOM bit.**

Note that, when other register accesses (reading of a CGMCTRL register is included) by software (interruption) or DMA are performed at this time, it is not regarded as continuation access but the forced shutdown request is invalid.

Figure 13-86. Error Handling

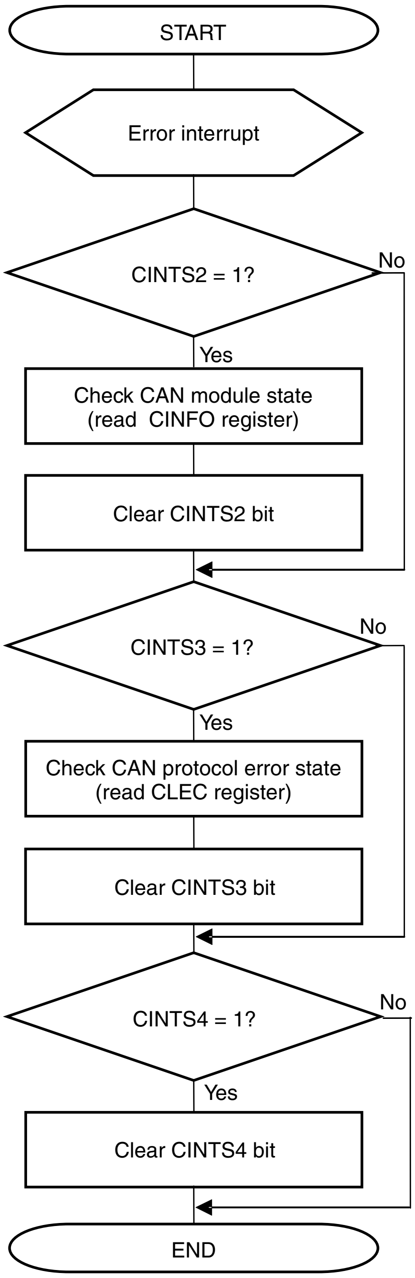
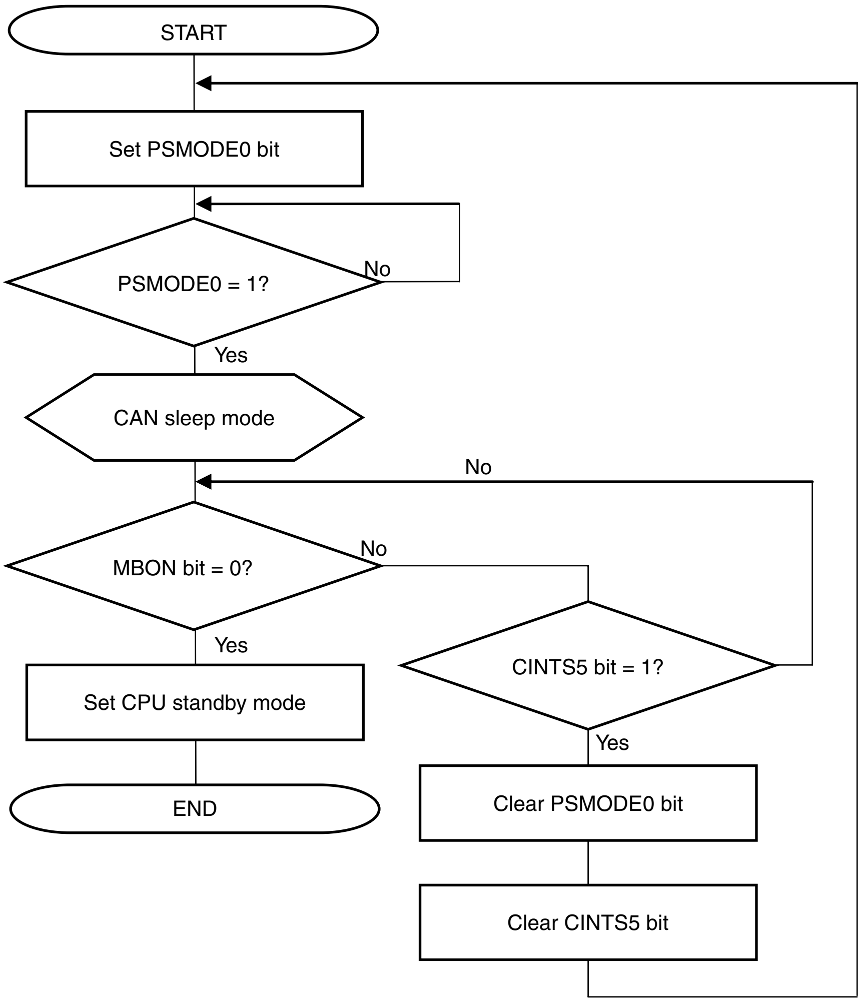
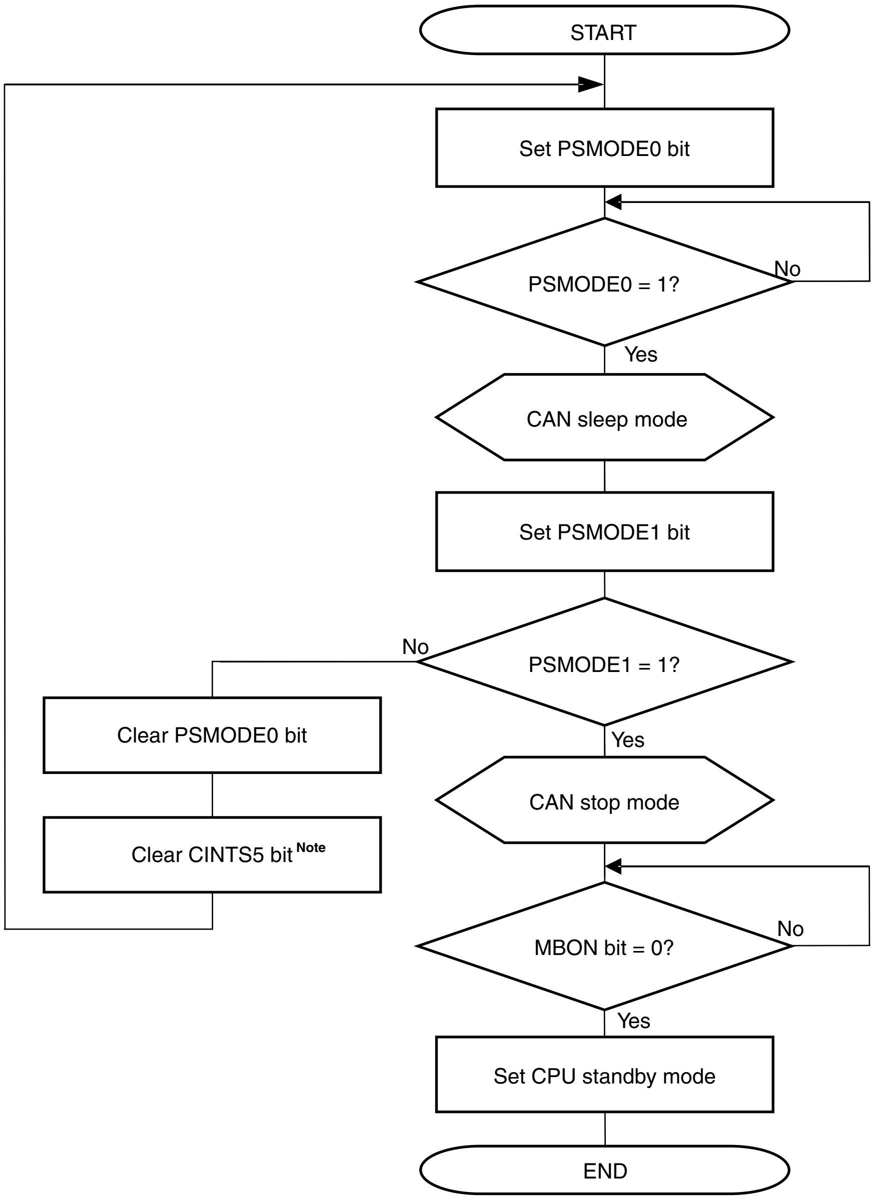


Figure 13-87. Setting CPU Standby (from CAN Sleep Mode)



Caution **Before the CPU is set in the CPU standby mode, please check the CAN sleep mode or not. However, after check of the CAN sleep mode, until the CPU is set in the CPU standby mode, the CAN sleep mode may be cancelled by wakeup from CAN bus.**

Figure 13-88. Setting CPU Standby (from CAN Stop Mode)



Note During wakeup interrupts

Caution The CAN stop mode can only be released by writing 01B to the PSMODE[1:0] bit of the CCTRL register and not by a change in the CAN bus state.

CHAPTER 14 MULTIPLIER/DIVIDER

14.1 Functions of Multiplier/Divider

The multiplier/divider has the following functions.

- 16 bits × 16 bits = 32 bits (multiplication)
- 32 bits ÷ 32 bits = 32 bits, 32-bit remainder (division)

14.2 Configuration of Multiplier/Divider

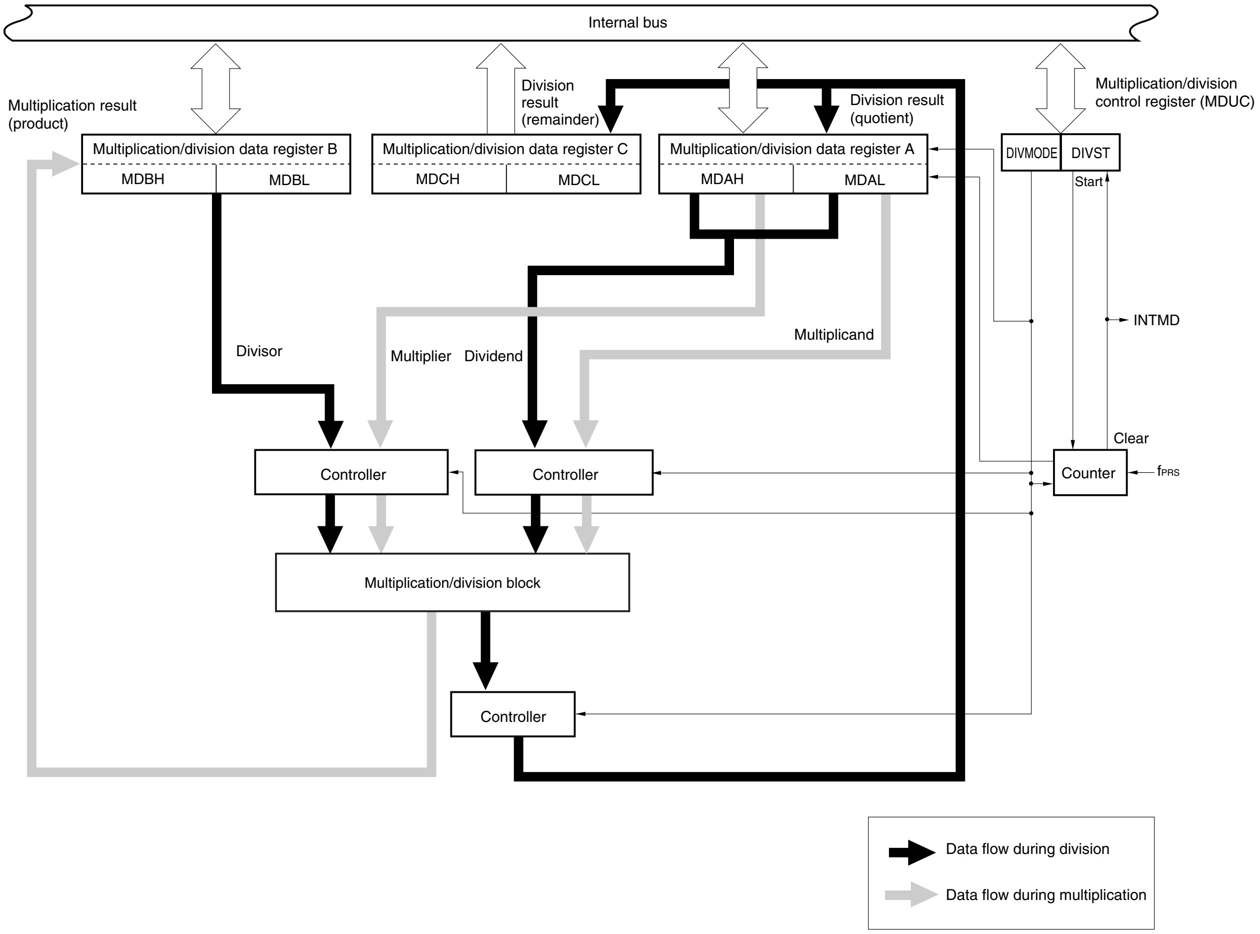
The multiplier/divider consists of the following hardware.

Table 14-1. 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 14-1 shows a block diagram of the multiplier/divider.

Figure 14-1. 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 14-2. 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 14-2. 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: Dividend (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 14-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)

Address: FFFF4H, FFFF5H, FFFF6H, FFFF7H After reset: 0000H, 0000H R/W

Symbol	FFFF5H								FFFF4H							
MDBH	MDBH 15	MDBH 14	MDBH 13	MDBH 12	MDBH 11	MDBH 10	MDBH 9	MDBH 8	MDBH 7	MDBH 6	MDBH 5	MDBH 4	MDBH 3	MDBH 2	MDBH 1	MDBH 0

Symbol	FFFF7H								FFFF6H							
MDBL	MDBL 15	MDBL 14	MDBL 13	MDBL 12	MDBL 11	MDBL 10	MDBL 9	MDBL 8	MDBL 7	MDBL 6	MDBL 5	MDBL 4	MDBHL 3	MDBL 2	MDBL 1	MDBL 0

Cautions 1. Do not rewrite the MDBH and MDBL 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 14-3. 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: Divisor (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 14-4. 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 14-4. 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)]

14.3 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 14-5. 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).

14.4 Operations of Multiplier/Divider

14.4.1 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).

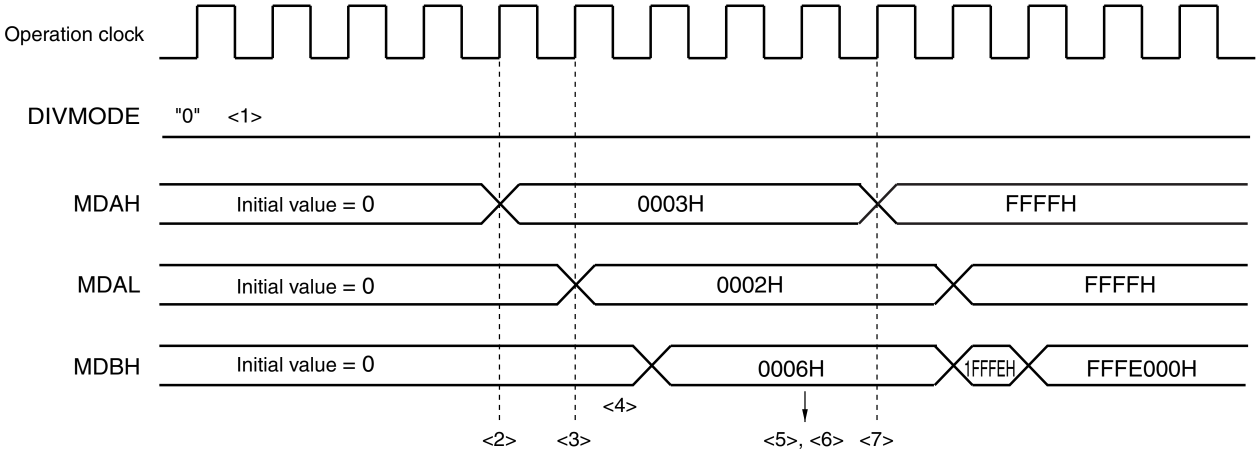
  (There is no preference in the order of executing steps <5> and <6>.)
- Next operation

  <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 **14.4.2 Division operation**.

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 14-6.

Figure 14-6. Timing Diagram of Multiplication Operation (0003H × 0002H)



14.4.2 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

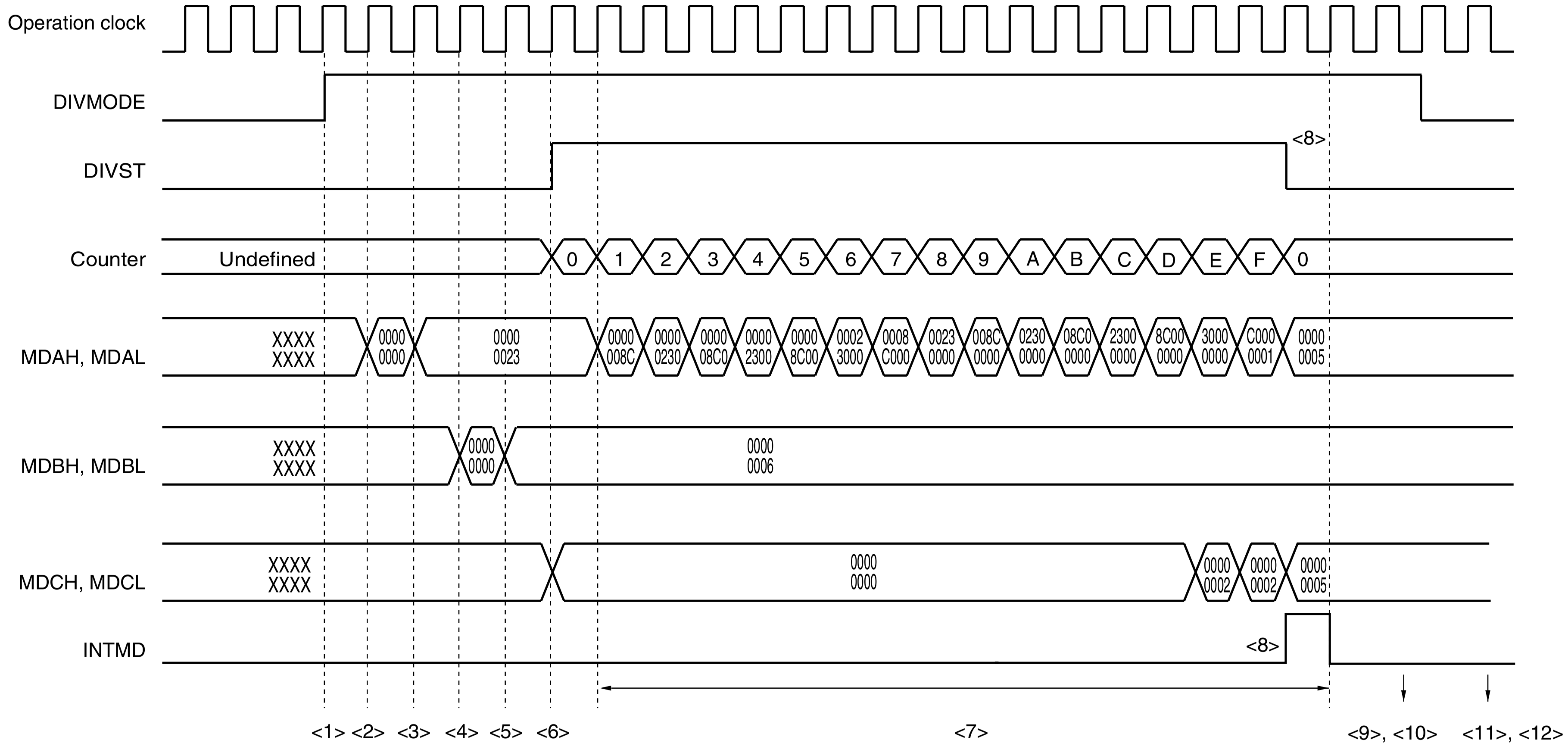
  <8> DIVST is cleared (0) and an interrupt request signal (INTMD) is generated (end of operation).
  <9> Read the quotient (lower 16 bits) from MDAL.
  <10> Read the quotient (higher 16 bits) from MDAH.
  <11> Read the remainder (lower 16 bits) from multiplication/division data register C (L) (MDCL).
  <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 **14.4.1 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 14-7.

Figure 14-7. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



CHAPTER 15 DMA CONTROLLER

The 78K0R/Hx3 has an internal DMA (Direct Memory Access) controller.

Data can be automatically transferred between the peripheral hardware supporting DMA, SFRs, and internal RAM without via CPU.

As a result, the normal internal operation of the CPU and data transfer can be executed in parallel with transfer between the SFR and internal RAM, and therefore, a large capacity of data can be processed. In addition, real-time control using communication, timer, and A/D can also be realized.

15.1 Functions of DMA Controller

○ Number of DMA channels: 4 channels
○ Transfer unit: 8 or 16 bits
○ Maximum transfer unit: 1024 times
○ Transfer type: 2-cycle transfer (One transfer is processed in 2 clocks and the CPU waits during that processing.)
○ Transfer mode: Single-transfer mode
○ Transfer request: Selectable from the following peripheral hardware interrupts
- A/D converter
- Serial interface (CIS00, CSI01, CSI10, CSI11, UART2, IIC11, IIC20)
- Timer (channel 1, 3, 5, or 7 of timer array unit 0, 1, channel 1, or 3 of timer array unit 2)
- LIN-UART
- Multiplier/divider

○ Transfer target: Between SFR and internal RAM

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

15.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

Table 15-1. Configuration of DMA Controller

Item	Configuration
Address registers	• DMA SFR address registers 0, 1 (DSA0, DSA1) • DMA RAM address registers 0, 1 (DRA0, DRA1)
Count register	• DMA byte count registers 0, 1 (DBC0, DBC1)
Control registers	• DMA mode control registers 0, 1 (DMC0, DMC1) • DMA operation control register 0, 1 (DRC0, DRC1) • DMA all-channel forced wait register (DMCALL)

Remark n: DMA channel number
n = 0 to 3

(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.

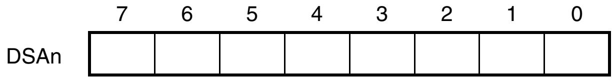
DSAn can be read or written in 8-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 00H.

Note Except for address FFFFEH because the PMC register is allocated there.

Figure 15-1. Format of DMA SFR Address Register n (DSAn)

Address: FFFB0H (DSA0), FFFB1H (DSA1) After reset: 00H R/W
F0080H (DSA2), F0081H (DSA3)



Remark n: DMA channel number
n = 0 to 3

(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 (FBF00H to FFEDFH in the case of the μPD78F1045, 78F1050) 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.

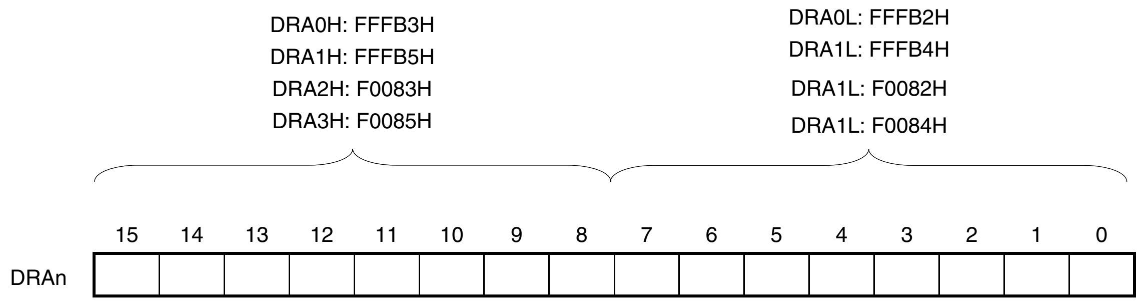
In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

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 15-2. Format of DMA RAM Address Register n (DRAn)

Address: FFFB2H, FFFB3H (DRA0), FFFB4H, FFFB5H (DRA1) After reset: 0000H R/W
F0082H, F0083H (DRA2), F0084H, F0085H (DRA3)



Remark n: DMA channel number
n = 0 to 3

(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 15-3. Format of DMA Byte Count Register n (DBCn)

Address: FFFB6H, FFFB7H (DBC0), FFFB8H, FFFB9H (DBC1) After reset: 0000H R/W
F0086H, F0087H (DBC2), F0088H, F0089H (DRA4)

DBC0H: FFFB7H
DBC1H: FFFB9H
DBC2H: F0087H
DBC3H: F0089H

DBC0L: FFFB6H
DBC1L: FFFB8H
DBC2L: F0086H
DBC3L: F0088H

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

DBCn[9:0]	Number of times of transfer (when DBCn is written)	Remaining number of times of transfer (when DBCn is read)
000H	1024	Completion of transfer or waiting for 1024 times of DMA transfer
001H	1	Waiting for remaining one time of DMA transfer
002H	2	Waiting for remaining two times of DMA transfer
003H	3	Waiting for remaining three times of DMA transfer
• • •	• • •	• • •
3FEH	1022	Waiting for remaining 1022 times of DMA transfer
3FFH	1023	Waiting for remaining 1023 times of DMA transfer

Cautions 1. Be sure to clear bits 15 to 10 to “0”.

2. If the general-purpose register is specified or the internal RAM space is exceeded as a result of continuous transfer, the general-purpose register or SFR space are written or read, resulting in loss of data in these spaces. Be sure to set the number of times of transfer that is within the internal RAM space.

Remark n: DMA channel number
n = 0 to 3

15.3 Registers to Controlling DMA Controller

DMA controller is controlled by the following registers.

- DMA mode control register n (DMCn)
- DMA operation control register n (DRCn)
- DMA all-channel forced wait register (DMCALL)

Remark n: DMA channel number
n = 0 to 3

(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, 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 15-4. Format of DMA Mode Control Register n (DMCn) (1/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W
F008AH (DMC3), F008BH (DMC3)

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	0	IFCn3	IFCn2	IFCn1	IFCn0

STGn[Note]	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).
DMA transfer is performed once by writing 1 to 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

Note The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 values.

Remark n: DMA channel number
n = 0 to 3

Figure 15-4. Format of DMA Mode Control Register n (DMCn) (2/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W
F008AH (DMC2), F008BH (DMC3)

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	0	IFCn3	IFCn2	IFCn1	IFCn0

IFCn 3	IFCn 2	IFCn 1	IFCn 0	Selection of DMAn start source[Note]			
				DMA0, DMA1		DMA2, DMA3	
				Trigger signal	Trigger contents	Trigger signal	Trigger contents
0	0	0	0	–	Disables DMA transfer by interrupt. (Only software trigger is enabled.)	–	Disables DMA transfer by interrupt. (Only software trigger is enabled.)
0	0	0	1	INTTM01	Timer channel 01 interrupt	INTTM15	Timer channel 15 interrupt
0	0	1	0	INTTM03	Timer channel 03 interrupt	INTTM17	Timer channel 17 interrupt
0	0	1	1	INTTM05	Timer channel 05 interrupt	INTTM21	Timer channel 21 interrupt
0	1	0	0	INTTM07	Timer channel 07 interrupt	INTTM23	Timer channel 23 interrupt
0	1	0	1	INTTM11	Timer channel 11 interrupt	INTCSI11/ INTIIC11	CSI11 transfer end interrupt/ IIC11 transfer end interrupt
0	1	1	0	INTTM13	Timer channel 13 interrupt	INTST2/ INTIIC20	UART2 transmission end interrupt/ IIC20 transfer end interrupt
0	1	1	1	INTCSI10	CSI10 transfer end interrupt	INTSR2	UART2 reception end interrupt
1	0	0	0	INTLT0	LIN-UART0 transmission interrupt	INTLT0	LIN-UART0 transmission interrupt
1	0	0	1	INTLR0	LIN-UART0 reception end interrupt	INTLR0	LIN-UART0 reception end interrupt
1	0	1	0	INTLT1	LIN-UART1 transmission interrupt	INTLT1	LIN-UART1 transmission interrupt
1	0	1	1	INTLR1	LIN-UART1 reception end interrupt	INTLR1	LIN-UART1 reception end interrupt
1	1	0	0	INTCSI00	CSI00 transfer end interrupt	INTCSI00	CSI00 transfer end interrupt
1	1	0	1	INTCSI01	CSI01 transfer end interrupt	INTCSI01	CSI01 transfer end interrupt
1	1	1	0	INTMD	Multiply/divide operation end interrupt	INTMD	Multiply/divide operation end interrupt
1	1	1	1	INTAD	A/D conversion end interrupt	INTAD	A/D conversion end interrupt

Note The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 values.

Caution The number of interrupts differs depending on the product. For details, see CHAPTER 16 INTERRUPT FUNCTIONS.

Remark n: DMA channel number
n = 0 to 3

(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 15-5. Format of DMA Operation Control Register n (DRCn)

Address: FFFBCH (DRC0), FFFBDH (DRC1) After reset: 00H R/W
F008CH (DRC2), F008DH (DRC3)

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 clock 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.	

Caution The DSTn flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DENn flag is enabled only when DSTn = 0. When a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, therefore, set DSTn to 0 and then DENn to 0 (for details, see 15.5.5 Forced termination by software).

Remark n: DMA channel number
n = 0 to 3

(3) DMA all-channel forced wait register (DMCALL)

This register is used to force DMA transfer on all channels to wait.

This register can also be used to change the priority order of the transfer channels.

Bit 7 (DRPMOD) of DMCALL can be rewritten while DMA transfer is in progress without affecting the current transfer.

Address: F008H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
DMCALL	DRPMOD	0	0	0	0	0	0	DWAITALL

DRPMOD	Transfer channel priority order
0	Priority order fixed (CH0 → CH1 → CH2 → CH3)
1	Priority order can be changed.

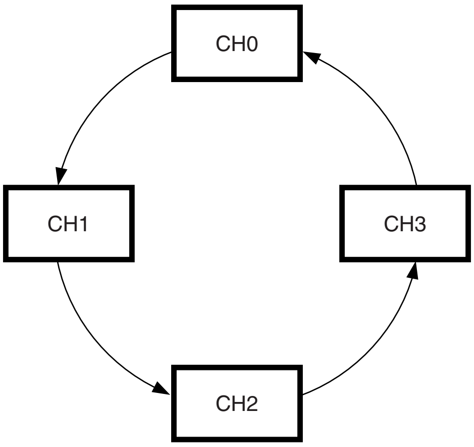
DWAITALL	All-channel forced wait
0	All channels are operating normally.
1	All channels are being forced to wait.

Remark If the order of priority is changed, the channel for which the number is equal to the current highest priority channel + 1 becomes the highest priority channel whenever a DMA transfer for which a request is received finishes, which results in constant rotation of the order of priority.

Regardless of whether there is a DMA request conflict, and regardless of which channel the request is for, the order of priority rotates each time a DMA transfer finishes. The initial value for the channel order of priority is CH0.

Example: If CH0 has the highest priority and a CH2 request is received, CH1 has the highest priority next.

If CH1 has the highest priority and a CH0 request is received, CH2 has the highest priority next.



15.4 Operation of DMA Controller

15.4.1 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, CBCn, and DMCn registers.

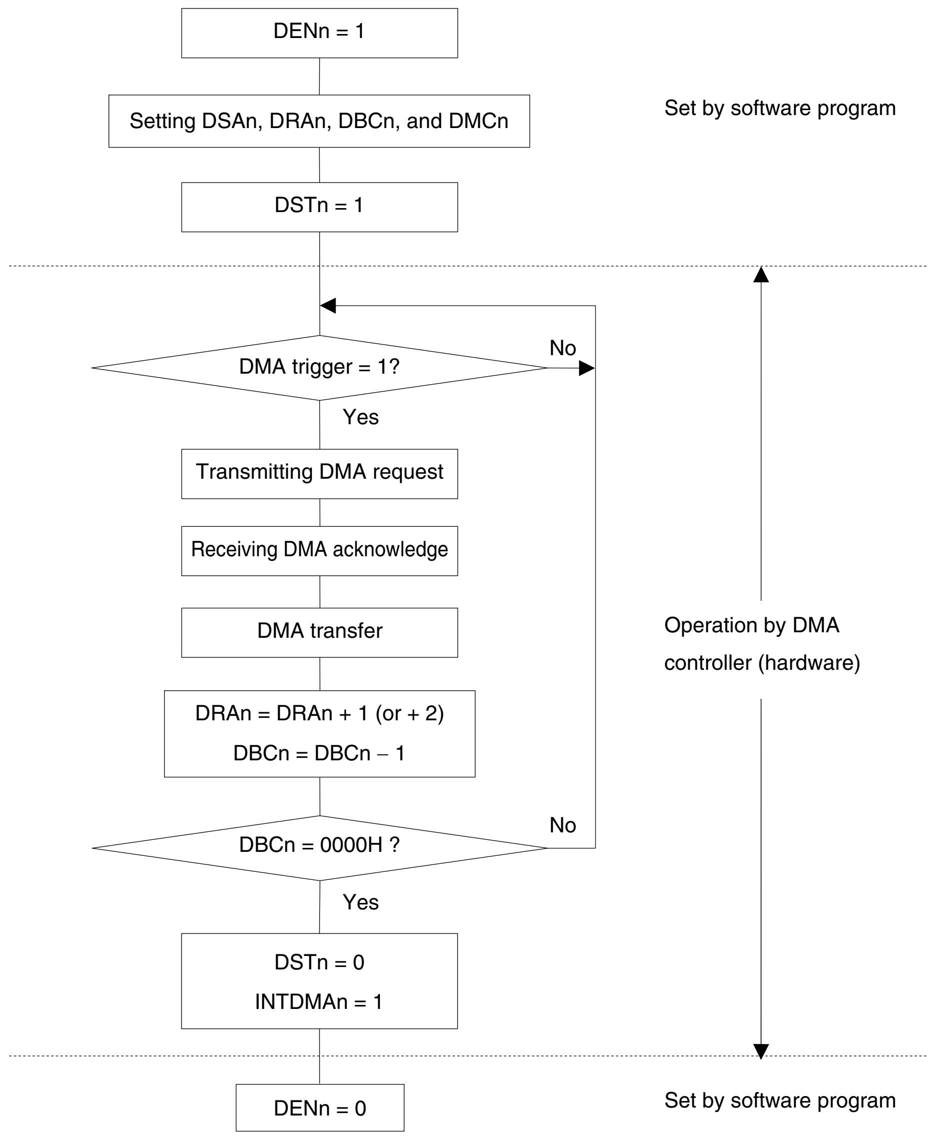
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.

<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).

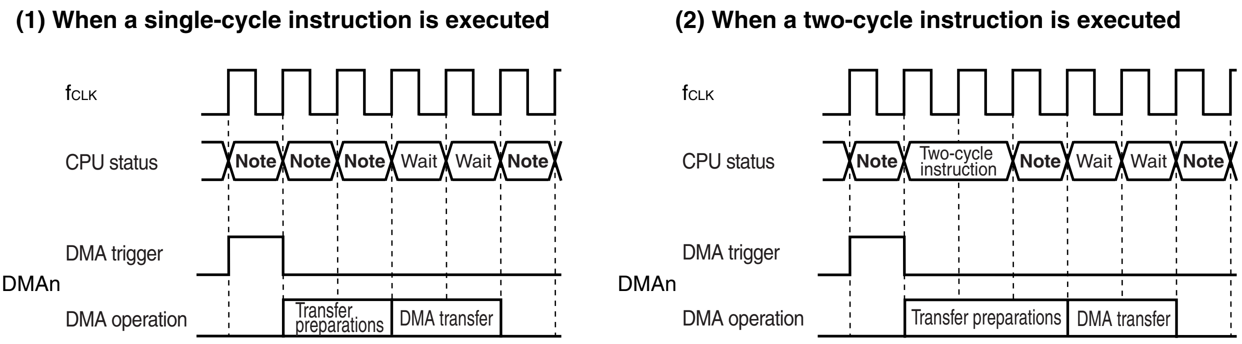
<6> Stop the operation of the DMA controller by clearing DENn to 0 when the DMA controller is not used.

Figure 15-6. Operation Procedure



Remark n: DMA channel number
n = 0 to 3

Figure 15-7. Operation Description


(1) When a single-cycle instruction is executed

(2) When a two-cycle instruction is executed

Note Single-cycle instruction

15.4.2 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.

15.4.3 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 to 3

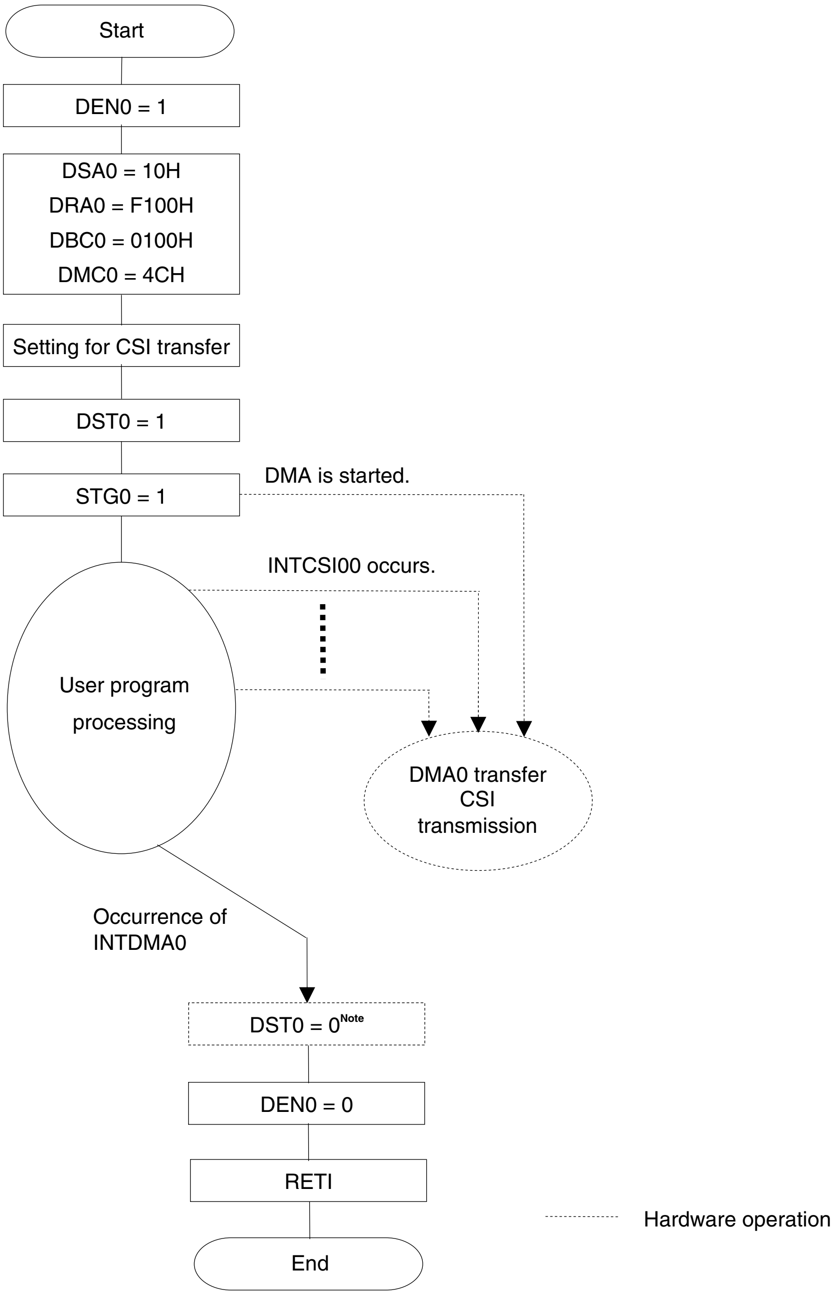
15.5 Example of Setting of DMA Controller

15.5.1 CSI consecutive transmission

A flowchart showing an example of setting for CSI consecutive transmission is shown below.

- Consecutive transmission of CSI00 (256 bytes)
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI00 (software trigger (STG0) only for the first start source)
- Interrupt of CSI00 is specified by IFC03 to IFC00 (bits 3 to 0 of the DMC0 register) = 1100B.
- Transfers FF100H to FF1FFH (256 bytes) of RAM to FFF10H of the data register (SDR00L) of CSI.

Figure 15-8. Example of Setting for CSI Consecutive Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set DST0 to 0 and then DEN0 to 0 (for details, see **15.5.5 Forced termination by software**).

The fist 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.

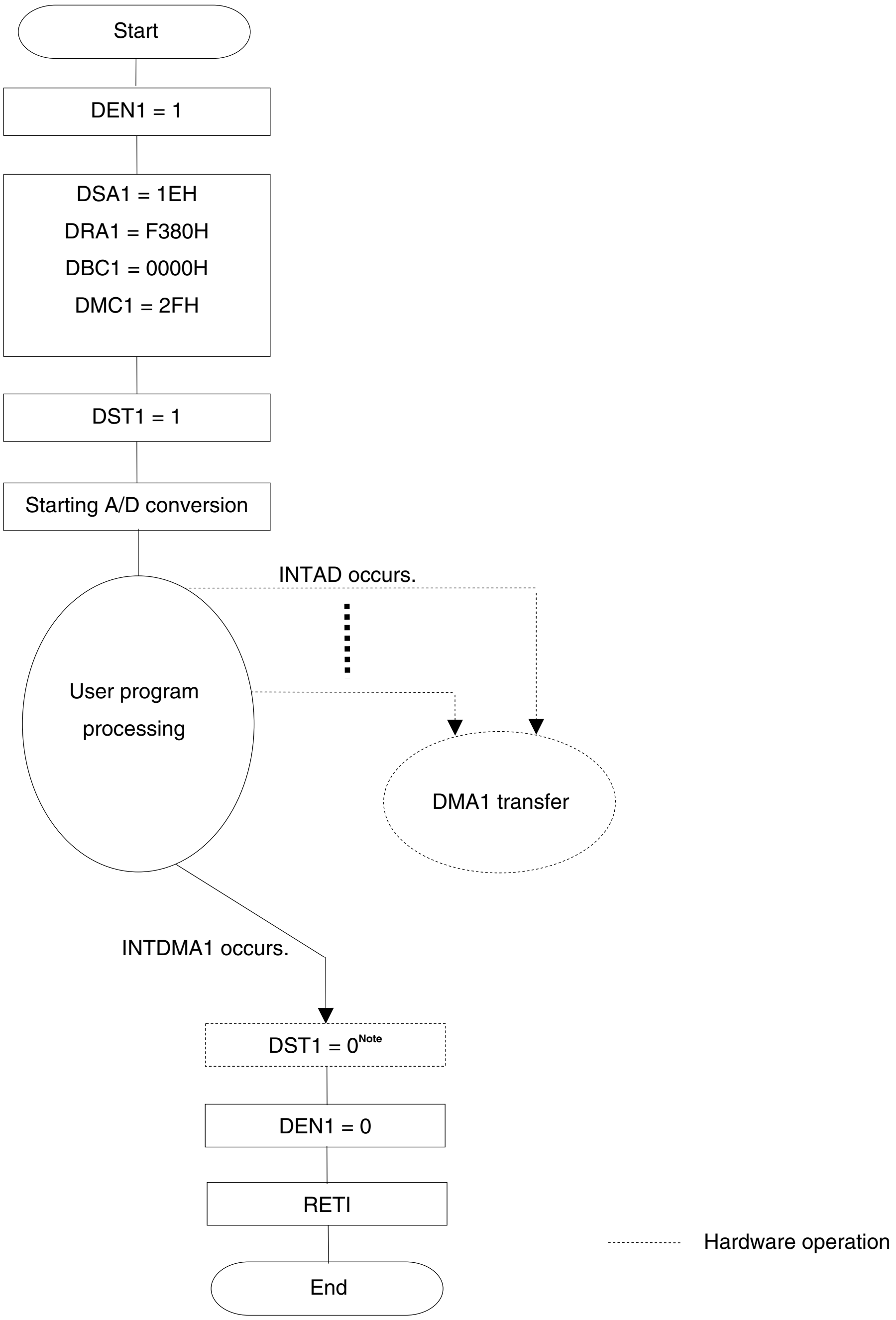
A DMA interrupt (INTDMA0) occurs when the last transmit data has been written to the data register.

15.5.2 Consecutive capturing of A/D conversion results

A flowchart of an example of setting for consecutively capturing A/D conversion results is shown below.

- Consecutive capturing of A/D conversion results.
- DMA channel 1 is used for DMA transfer.
- DMA start source: INTAD
- Interrupt of A/D is specified by IFC13 to IFC10 (bits 3 to 0 of the DMC1 register) = 1111B.
- Transfers FFF1EH and FFF1FH (2 bytes) of the 10-bit A/D conversion result register to 2048 bytes of FF380H to FFB7FH of RAM.

Figure 15-9. Example of Setting of Consecutively Capturing A/D Conversion Results



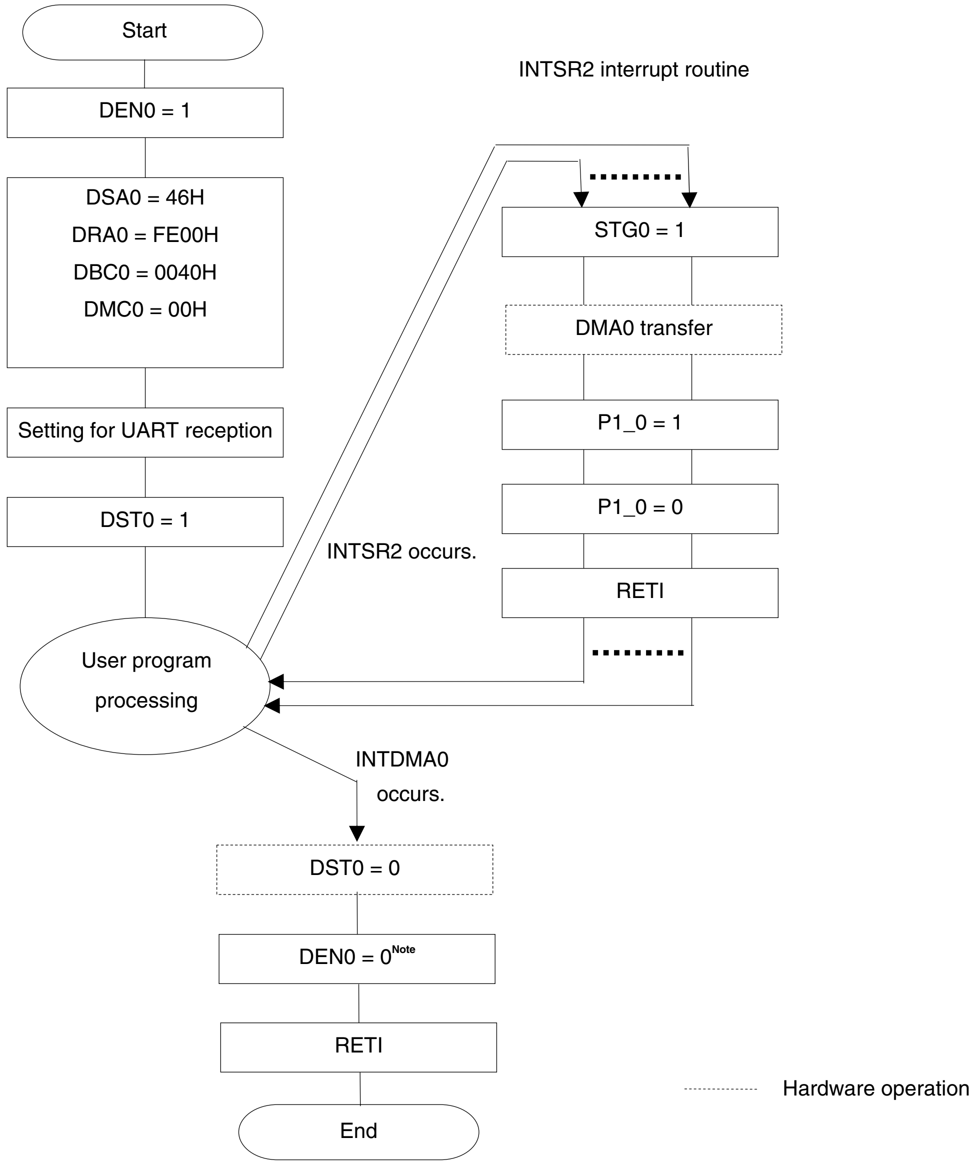
Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set DST1 to 0 and then DEN1 to 0 (for details, see **15.5.5 Forced termination by software**).

15.5.3 UART consecutive reception + ACK transmission

A flowchart illustrating an example of setting for UART consecutive reception + ACK transmission is shown below.

- Consecutively receives data from UART2 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 FFF46H of UART receive data register 2 (RXD2) to 64 bytes of FFE00H to FFE3FH of RAM.

Figure 15-10. 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, see **15.5.5 Forced termination by software**).

Remark This is an example where a software trigger is used as a DMA start source.
If ACK is not transmitted and if only data is consecutively received from UART, the UART2reception end interrupt (INTSR2) can be used to start DMA for data reception.

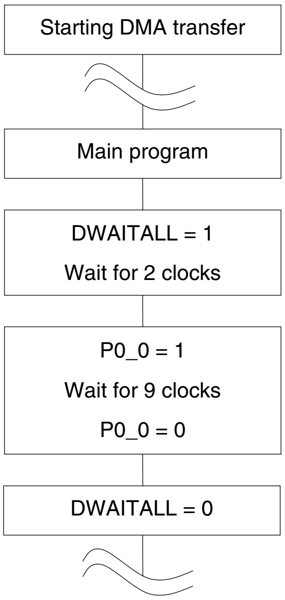
15.5.4 Holding DMA transfer pending by DWAITALL

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 DWAITALL to 1.

To output a pulse with a width of 10 clocks of the operating frequency from the P00 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 DWAITALL to 1.

After setting DWAITALL to 1, it takes two clocks until a DMA transfer is held pending.

Figure 15-11. Example of Setting for Holding DMA Transfer Pending by DWAITALL



Caution Even if a transfer trigger is issued more than once for the same channel when the transfer of that channel is being held pending, only one transfer will be executed after the pending status is cancelled.

Remarks 1. n: DMA channel number
n = 0 to 3

2. 1 clock: 1/fCLK (fCLK: CPU clock)

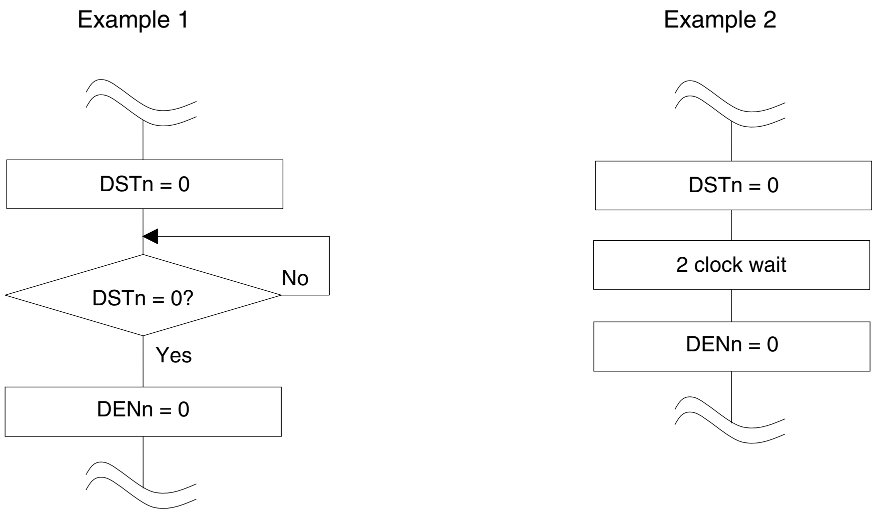
15.5.5 Forced termination by software

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.

- Set DSTn to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software, confirm by polling that DSTn has actually been cleared to 0, and then set DENn to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction).

- Set DSTn to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software and then set DENn to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction) two or more clocks after.

Figure 15-12. Forced Termination of DMA Transfer



Remarks 1. n: DMA channel number
n = 0 to 3

2. 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

15.6 Cautions on Using DMA Controller

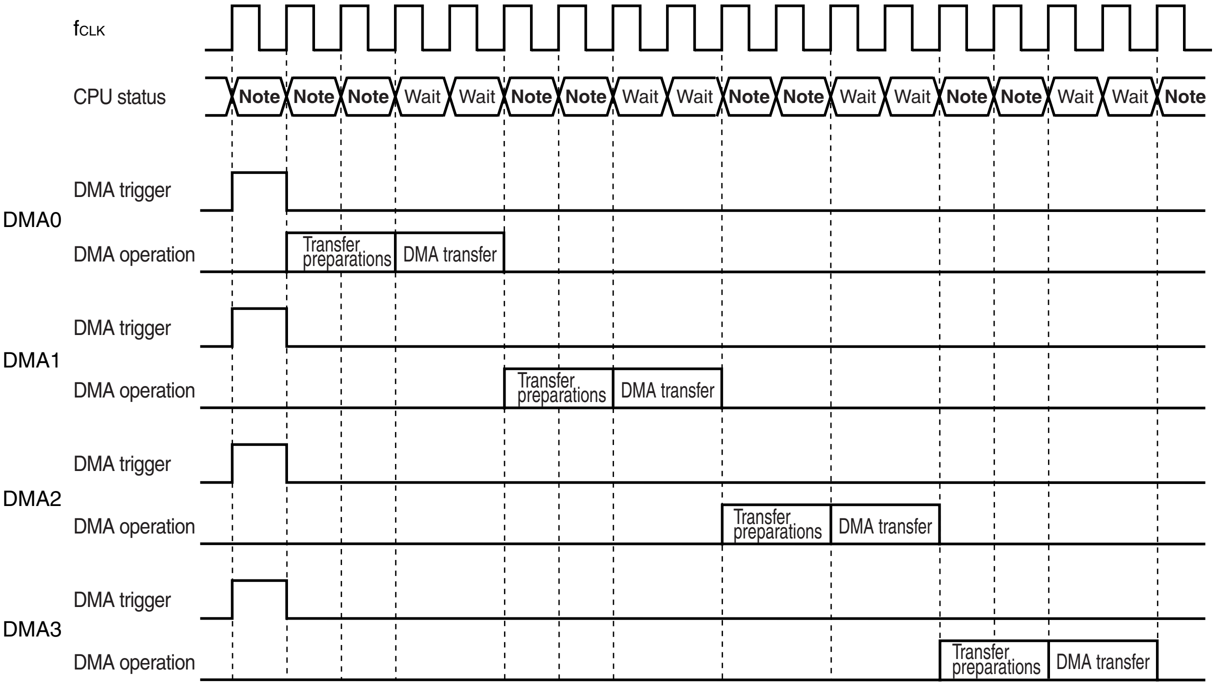
(1) Priority of DMA

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.

However, if a DMA request is generated at the same time, the order of priority specified by bit 7 (DRPMOD) of the DMCALL register is used. Here, *the same time* refers to the period of time from when the DMA request is generated until one clock before the DMA transfer starts.

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.

Figure 15-13. Operation Example When DMA Request Triggers of Each DMA Channel Conflict (If Order of Priority Is CH0, CH1, CH2, and Then CH3)



Note Single-cycle instruction

(2) DMA response time

The response time of DMA transfer is as follows.

Table 15-2. Response Time of DMA Transfer

	Minimum Time	Maximum Time
Response time[Note]	3 clocks	10 clocks

Note This is the time required to execute an instruction from internal ROM. However, DMA transfers might be further delayed in the cases below.

- When executing an instruction from the internal RAM
- When accessing the data flash memory
- When accessing an SFR for which waiting is necessary (a CAN controller register other than a control register (the peripheral clock select register (PCKSEL), serial communication pin select register (STSEL), port registers 1 and 7 (P1, P7), and port mode registers 1 and 7 (PM1, PM7))). (For details, see **Table 13-15**.)
- When executing a DMA hold instruction
- When generating another DMA request

Cautions 1. The above response time does not include the two clock cycles required for a DMA transfer.

2. 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.

3. When using the EEPROM emulation library for the data flash memory, DMA transfers might be delayed even when not accessing the data flash memory.

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 15-3. 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, IF3L, IF3H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H and PSW each.

(5) Operation when instruction to access SFR requiring wait is executed[Note]

If the instruction to access the SFR register requiring wait is executed, DMA transfer will be held pending.
If polling of the SFR register requiring wait is continued, DMA transfer will be held pending continuously.

Note The CAN controller registers are the SFRs for which waiting is necessary. However, the control registers (the peripheral clock select register (PCKSEL), serial communication pin select register (STSEL), port registers 1 and 7 (P1, P7), and port mode registers 1 and 7 (PM1, PM7)) are excluded. (For details, see **Table 13-15**.)

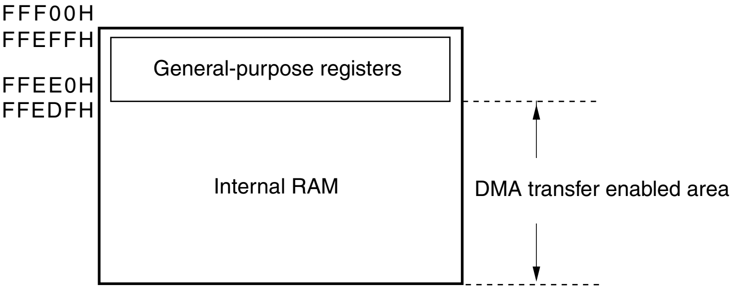
(6) Operation if address in general-purpose register area or other than those of internal RAM area is specified

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 16 INTERRUPT FUNCTIONS

		78K0R/HC3 (μPD78F10yy)	78K0R/HE3 (μPD78F10yy)	78K0R/HF3 (μPD78F10yy)	78K0R/HG3 (μPD78F10yy)
		yy = 31 to 35	yy = 36 to 40	yy = 41 to 45	yy = 46 to 50
Maskable interrupts	External	10	11	12	12
	Internal	40	47	47	49

16.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H).

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 16-1**.

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.

<R> 16.2 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 16-1**). 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 16-1. Interrupt Source List (1/3)

Interrupt Type	Internal/ External	Basic Configuration Type[Note 1]	Default Priority[Note 2]	Interrupt Source		Vector Table Address	HC3	HE3	HF3	HG3
				Name	Trigger					
Maskable	Internal	(A)	0	INTWDTI	Watchdog timer interval[Note 3] (75% of overflow time)	0004H	√	√	√	√
			1	INTLVI	Low-voltage detection[Note 4]	0006H	√	√	√	√
	External	(B)	2	INTP0	Pin input edge detection	0008H	√	√	√	√
			3	INTP1		000AH	√	√	√	√
			4	INTP2		000CH	√	√	√	√
			5	INTP3		000EH	√	√	√	√
			6	INTP4		0010H	√	√	√	√
			7	INTP5		0012H	√	√	√	√
	Internal	(A)	8	INTCLM	To stop PLL clock	0014H	√	√	√	√
			9	INTCSI00	End of CSI00 communication	0016H	√	√	√	√
			10	INTCSI01	End of CSI01 communication	0018H	–	√	√	√
			11	INTDMA0	End of DMA0 transfer	001AH	√	√	√	√
			12	INTDMA1	End of DMA1 transfer	001CH	√	√	√	√
			13	INTWUTM	Wakeup timer compare match	001EH	√	√	√	√
			14	INTFL	End of data flash programming	0020H	√	√	√	√
			15	INTLT0	Start of LIN-UART0 transmission or transmission completion	0022H	√	√	√	√
			16	INTLR0	LIN-UART0 reception completion	0024H	√	√	√	√
			17	INTLS0	LIN-UART0 reception status error	0026H	√	√	√	√
	External	(B)	18	INTPLR0	LIN-UART0 reception pin input	0028H	√	√	√	√
			19	INTP8	Pin input edge detection	002AH	–	–	√	√
	Internal	(A)	20	INTTM00	End of timer array unit 00 count or capture	002CH	√	√	√	√
			21	INTTM01	End of timer array unit 01 count or capture	002FH	√	√	√	√
			22	INTTM02	End of timer array unit 02 count or capture	0030H	√	√	√	√

Notes 1. Basic configuration types (A) to (D) correspond to (A) to (D) in **Figure 16-1**.

2. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 60 indicates the lowest priority.

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 16-1. Interrupt Source List (2/3)

Interrupt Type	Internal/ External	Basic Configuration Type[Note 1]	Default Priority[Note 2]	Interrupt Source		Vector Table Address	HC3	HE3	HF3	HG3
				Name	Trigger					
Maskable	Internal	(A)	23	INTTM03	End of timer array unit 03 count or capture	0032H	√	√	√	√
			24	INTAD	End of A/D conversion	0034H	√	√	√	√
			25	INTLT1	Start of LIN-UART1 transmission or transmission completion	0036H	√	√	√	√
			26	INTLR1	LIN-UART1 reception completion	0038H	√	√	√	√
			27	INTLS1	LIN-UART1 reception status error	003AH	√	√	√	√
	External	(B)	28	INTPLR1	LIN-UART1 reception pin input	003CH	√	√	√	√
	Internal	(A)	29	INTCSI10	End of CSI10 communication	003EH	√	√	√	√
			30	INTCSI11/ INTIIC11	End of CSI11 communication/ end of IIC11 communication	0040H	–	–	–	√
							√	√	√	√
			31	INTTM04	End of timer array unit 04 count or capture	0042H	√	√	√	√
			32	INTTM05	End of timer array unit 05 count or capture	0044H	√	√	√	√
			33	INTTM06	End of timer array unit 06 count or capture	0046H	√	√	√	√
			34	INTTM07	End of timer array unit 07 count or capture	0048H	√	√	√	√
	External	(B)	35	INTP6	Pin input edge detection	004AH	√	√	√	√
		(C)		INTKR	key return signal detection					
		(B)	36	INTP7	Pin input edge detection	004CH	√	√	√	√
	Internal	(A)	37	INTC0ERR	CAN error	004EH	√	√	√	√
			38	INTC0WUP	CAN wakeup	0050H	√	√	√	√
			39	INTC0REC	CAN reception completion	0052H	√	√	√	√
			40	INTC0TRX	CAN transmission completion	0054H	√	√	√	√
			41	INTTM10	End of timer channel 10 count or capture	0056H	√	√	√	√
			42	INTTM11	End of timer channel 11 count or capture	0058H	√	√	√	√
			43	INTTM12	End of timer channel 12 count or capture	005AH	√	√	√	√
			44	INTTM13	End of timer channel 13 count or capture	005CH	√	√	√	√
			45	INTMD	End of division operation	005EH	√	√	√	√
			46	INTST2/ INTIIC20	End of UART2 transmission/ end of IIC20 communication	0060H	–	√	√	√
			47	INTSR2	End of UART2 reception	0062H	–	√	√	√

Notes 1. Basic configuration types (A) to (D) correspond to (A) to (D) in **Figure 16-1**.

2. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 60 indicates the lowest priority.

Table 16-1. Interrupt Source List (3/3)

Interrupt Type	Internal/ External	Basic Configuration Type[Note 1]	Default Priority[Note 2]	Interrupt Source		Vector Table Address	HC3	HE3	HF3	HG3
				Name	Trigger					
Maskable	External	(B)	48	INTPR2	UART2 reception pin input	0064H	–	√	√	√
	Internal	(A)	49	INTTM14	End of timer channel 14 count or capture	0066H	√	√	√	√
			50	INTTM15	End of timer channel 15 count or capture	0068H	√	√	√	√
			51	INTTM16	End of timer channel 16 count or capture	006AH	√	√	√	√
			52	INTTM17	End of timer channel 17 count or capture	006CH	√	√	√	√
			53	INTTM20	End of timer channel 20 count or capture	006EH	–	√	√	√
			54	INTTM21	End of timer channel 21 count or capture	0070H	–	√	√	√
			55	INTTM22	End of timer channel 22 count or capture	0072H	–	√	√	√
			56	INTTM23	End of timer channel 23 count or capture	0074H	–	√	√	√
			57	INTTM25	End of timer channel 25 count or capture	0076H	–	–	–	√
			58	INTTM27	End of timer channel 27 count or capture	0078H	–	–	–	√
			59	INTDMA2	End of DMA2 transfer	007AH	√	√	√	√
			60	INTDMA3	End of DMA3 transfer	007CH	√	√	√	√
Software	–	(D)	–	BRK	Execution of BRK instruction	007EH	√	√	√	√
Reset	–	–	–	RESET	$\overline{\text{RESET}}$ pin input	0000H	√	√	√	√
				POC	Power-on clear					
				LVI	Low-voltage detection[Note 3]					
				WDT	WDT overflow					
				TRAP	Execution of illegal instruction[Note 4]					
				IAW	illegal memory access					
				CLKM	Clock monitor					

Notes 1. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 16-1.

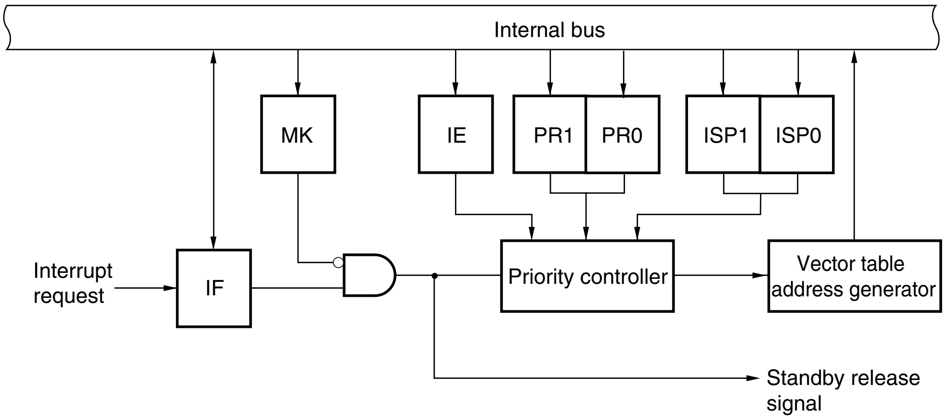
2. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 60 indicates the lowest priority.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

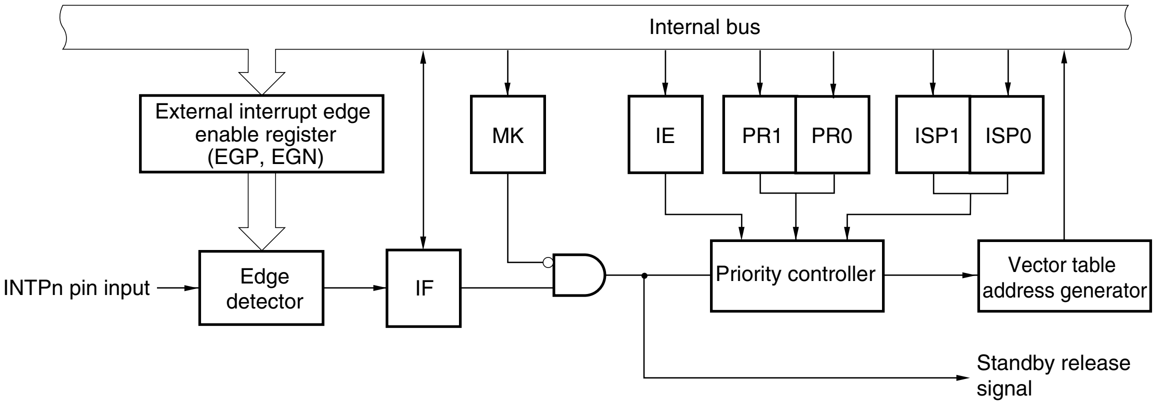
4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 16-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn, INTPR2, INTPLR0, INTPLR1)

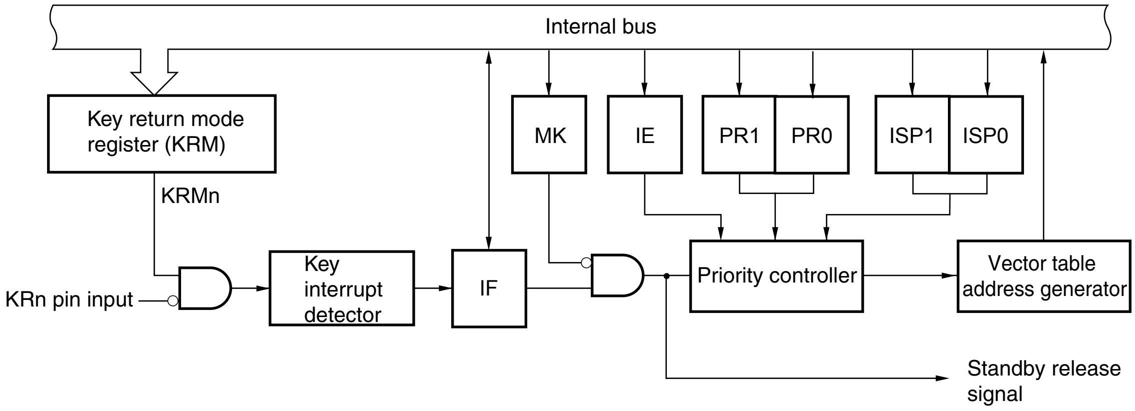


Remark n =0 to 7: 78K0R/HC3, 78K0R/HE3
n = 0 to 8: 78K0R/HF3, 78K0R/HG3

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

Figure 16-1. Basic Configuration of Interrupt Function (2/2)

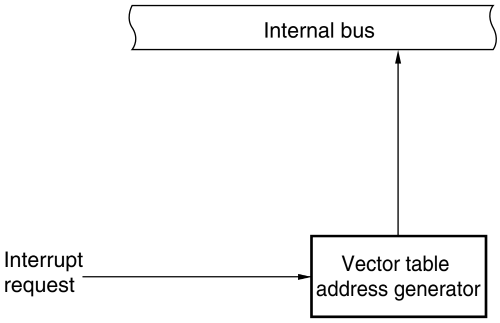
(C) External maskable interrupt (INTKR)



Remark n = 0 to 3: 78K0R/HC3

n = 0 to 7: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3

(D) Software interrupt



IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP0:	In-service priority flag 0
ISP1:	In-service priority flag 1
MK:	Interrupt mask flag
PR0:	Priority specification flag 0
PR1:	Priority specification flag 1

<R> 16.3 Registers Controlling Interrupt Functions

The following 7 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H)
- External interrupt rising edge enable registers (EGP0, EGP1)
- External interrupt falling edge enable registers (EGN0, EGN1)
- External interrupt input pin selection register 0 (IPSEL0)
- Program status word (PSW)

Table 16-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 16-2. Flags Corresponding to Interrupt Request Sources (1/4)

HC3	HE3	HF3	HG3	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	
√	√	√	√	INTP1	PIF1		PMK1		PPR01, PPR11	
√	√	√	√	INTP2	PIF2		PMK2		PPR02, PPR12	
√	√	√	√	INTP3	PIF3		PMK3		PPR03, PPR13	
√	√	√	√	INTP4	PIF4		PMK4		PPR04, PPR14	
√	√	√	√	INTP5	PIF5		PMK5		PPR05, PPR15	
√	√	√	√	INTCLM	CLMIF	IF0H	CLMMK	MK0H	CLMPR0, CLMPR1	PR00H, PR10H
√	√	√	√	INTCSI00	CSIIF00		CSIMK00		CSIPR000, CSIPR100	
–	√	√	√	INTCSI01	CSIIF01		CSIMK01		CSIPR001, CSIPR101	
√	√	√	√	INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10	
√	√	√	√	INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11	
√	√	√	√	INTWUTM	WUTMIF		WUTMMK		WUTMPR0, WUTMPR1	
√	√	√	√	INTFL	FLIF		FLMK		FLPR0, FLPR1	
√	√	√	√	INTLT0	LTIF0		LTMK0		LTPR00, LTPR10	

Remark √: Mounted

Table 16-2. Flags Corresponding to Interrupt Request Sources (2/4)

HC3	HE3	HF3	HG3	Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
						Register		Register		Register
√	√	√	√	INTLR0	LRIF0	IF1L	LRMK0	MK1L	LRPR00, LRPR10	PR01L, PR11L
√	√	√	√	INTLS0	LSIF0		LSMK0		LSPR00, LSPR10	
√	√	√	√	INTPLR0	PIFLR0		PMKLR0		PPR0LR0, PPR1LR0	
–	–	√	√	INTP8	PIF8		PMK8		PPR08, PPR18	
√	√	√	√	INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
√	√	√	√	INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
√	√	√	√	INTTM02	TMIF02		TMMK02		TMPR002, TMPR102	
√	√	√	√	INTTM03	TMIF03		TMMK03		TMPR003, TMPR103	
√	√	√	√	INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H
√	√	√	√	INTLT1	LTIF1		LTMK1		LTPR01, LTPR11	
√	√	√	√	INTLR1	LRIF1		LRMK1		LRPR01, LRPR11	
√	√	√	√	INTLS1	LSIF1		LSMK1		LSPR01, LSPR11	
√	√	√	√	INTPLR1	PIFLR1		PMKLR1		PPR0LR1, PPR1LR1	
√	√	√	√	INTCSI10	CSIIF10		CSIMK10		CSIPR010, CSIPR110	
–	–	–	√	INTCSI11[Note]	CSIIF11[Note]		CSIMK11[Note]		CSIPR011, CSIPR111[Note]	
√	√	√	√	INTIIC11[Note]	IICIF11[Note]		IICMK11[Note]		IICPR011, IICPR111[Note]	
√	√	√	√	INTTM04	TMIF04		TMMK04		TMPR004, TMPR104	

Note Do not use INTCSI11 and INTIIC11 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTCSI11 and INTIIC11 is generated, bit 6 of IF1H is set to 1. Bit 6 of MK1H, PR01H, and PR11H supports these three interrupt sources.

Remark √: Mounted, –: Not mounted

Table 16-2. Flags Corresponding to Interrupt Request Sources (3/4)

HC3	HE3	HF3	HG3	Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
						Register		Register		Register
√	√	√	√	INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L
√	√	√	√	INTTM06	TMIF06		TMMK06		TMPR006, TMPR106	
√	√	√	√	INTTM07	TMIF07		TMMK07		TMPR007, TMPR107	
√	√	√	√	INTP6[Note]	PIF6[Note]		PMK6[Note]		PPR06, PPR16[Note]	
√	√	√	√	INTKR[Note]	KRIF[Note]		KRMK[Note]		KRPR0, KRPR1[Note]	
√	√	√	√	INTP7	PIF7		PMK7		PPR07, PPR17	
√	√	√	√	INTC0ERR	C0ERRIF		C0ERRMK		C0ERRPR0, C0ERRPR1	
√	√	√	√	INTC0WUP	C0WUPIF		C0WUPMK		C0WUPPR0, C0WUPPR1	
√	√	√	√	INTC0REC	C0RECIF		C0RECMK		C0RECPR0, C0RECPR1	
√	√	√	√	INTC0TRX	C0TRXIF	IF2H	C0TRXMK	MK2H	C0TRXPR0, C0TRXPR1	PR02H, PR12H
√	√	√	√	INTTM10	TMIF10		TMMK10		TMPR010, TMPR110	
√	√	√	√	INTTM11	TMIF11		TMMK11		TMPR011, TMPR111	
√	√	√	√	INTTM12	TMIF12		TMMK12		TMPR012, TMPR112	
√	√	√	√	INTTM13	TMIF13		TMMK13		TMPR013, TMPR113	
√	√	√	√	INTMD	MDIF		MDMK		MDPR0, MDPR1	
–	√	√	√	INTST2[Note]	STIF2[Note]		STMK2[Note]		STPR02, STPR12[Note]	
–	√	√	√	INTIIC20[Note]	IICIF20[Note]		IICMK20[Note]		IICPR020, IICPR120[Note]	
–	√	√	√	INTSR2	SRIF2		SRMK2		SRPR02, SRPR12	

Note Do not use INTP6 and INTKR at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTP6 and INTKR is generated, bit 3 of IF2L is set to 1. Bit 3 of MK2L, PR02L, and PR12L supports these three interrupt sources.

Remark √: Mounted, –: Not mounted

Table 16-2. Flags Corresponding to Interrupt Request Sources (4/4)

HC3	HE3	HF3	HG3	Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
						Register		Register		Register
–	√	√	√	INTPR2	PIFR2	IF3L	PMKR2	MK3L	PPR0R2, PPR1R2	PR03L, PR13L
√	√	√	√	INTTM14	TMIF14		TMMK14		TMPR014, TMPR114	
√	√	√	√	INTTM15	TMIF15		TMMK15		TMPR015, TMPR115	
√	√	√	√	INTTM16	TMIF16		TMMK16		TMPR016, TMPR116	
√	√	√	√	INTTM17	TMIF17		TMMK17		TMPR017, TMPR117	
–	√	√	√	INTTM20	TMIF20		TMMK20		TMPR020, TMPR120	
–	√	√	√	INTTM21	TMIF21		TMMK21		TMPR021, TMPR121	
–	√	√	√	INTTM22	TMIF22		TMMK22		TMPR022, TMPR122	
–	√	√	√	INTTM23	TMIF23	IF3H	TMMK23	MK3H	TMPR023, TMPR123	PR03H, PR13H
–	–	–	√	INTTM25	TMIF25		TMMK25		TMPR025, TMPR125	
–	–	–	√	INTTM27	TMIF27		TMMK27		TMPR027, TMPR127	
√	√	√	√	INTDMA2	DMAIF2		DMAMK2		DMAPR02, DMAPR12	
√	√	√	√	INTDMA3	DMAIF3		DMAMK3		DMAPR03, DMAPR13	

Remark √: Mounted, –: Not mounted

(1) Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H)

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, IF2H, IF3L, and IF3H can be set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H, IF1L and IF1H, IF2L and IF2H, and IF3L and IF3H are combined to form 16-bit registers IF0, IF1, IF2, and IF3 they can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 16-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (78K0R/HG3) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	LTIF0	FLIF	WUTMIF	DMAIF1	DMAIF0	CSIIF01	CSIIF00	CLMIF

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	PIE8	PIELR00	LSIF0	LRIF0

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	TMIF04	CSIIF11 IICIF11	CSIIF10	PIFLR1	LSIF1	LRIF1	LTIF1	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	C0RECIF	C0WUPIF	C0ERRIF	PIF7	PIF6 KRIF	TMIF07	TMIF06	TMIF05

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	<6	<5>	<4>	<3>	<2>	<1>	<0>
IF2H	SRIF2	STIF2 IICIF20	MDIF	TMIF13	TMIF12	TMIF11	TMIF10	C0TRXIF

Figure 16-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (78K0R/HG3) (2/2)

Address: FFFD2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF3L	TMIF22	TMIF21	TMIF20	TMIF17	TMIF16	TMIF15	TMIF14	PIFR2

Address: FFFD3H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
IF3H	0	0	0	DMAIF3	DMAIF2	TMIF27	TMIF25	TMIF23

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. The above is the bit layout for the 78K0R/HG3. The available bits differ depending on the product. For details about the bits available for each product, see Table 16-2. Be sure to clear bits that are not available to 0.

2. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

3. 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.

4. The bits mounted depend on the product. See Table 16-2 Flags Corresponding to Interrupt Request Sources.

<R> **(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H)**

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.
MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, and MK3H can be set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, MK1L and MK1H, and MK2L and MK2H, and MK3L and MK3H are combined to form 16-bit registers MK0, MK1, MK2, and MK3, 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 16-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) (78K0R/HG3) (1/2)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	LTMK0	FLMK	WUTMMK	DMAMK1	DMAMK0	CSIMK01	CSIMK00	CLMMK

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	PMK8	PMKLR0	LSMK0	LRMK0

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	TMMK04	CSIMK11 IICMK11	CSIMK10	PMKLR1	LSMK1	LRMK1	LTMK1	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	C0RECMK	C0WUPMK	C0ERRMK	PMK7	PMK6 KRMK	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	SRMK2	STMK2 IICMK20	MDMK	TMMK13	TMMK12	TMMK11	TMMK10	C0TRXMK

Address: FFFD6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK3L	TMMK22	TMMK21	TMMK20	TMMK17	TMMK16	TMMK15	TMMK14	PMKR2

Figure 16-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) (78K0R/HG3) (2/2)

Address: FFFD7H After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
MK3H	1	1	1	DMAMK3	DMAMK2	TMMK27	TMMK25	TMMK23

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution The above is the bit layout for the 78K0R/HG3. The available bits differ depending on the product. For details about the bits available for each product, see Table 16-2. Be sure to set bits that are not available to 1.

<R> **(3) Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H)**

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, 2H, 3L, or 3H).

PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H can be set by a 1-bit or 8-bit memory manipulation instruction. If PR00L and PR00H, PR01L and PR01H, PR02L and PR02H, PR03L and PR03H, PR10L and PR10H, PR11L and PR11H, PR12L and PR12H, and PR13L and PR13H are combined to form 16-bit registers PR00, PR01, PR02, PR03, PR10, PR11, PR12, and PR13, 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 16-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (78K0R/HG3) (1/3)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	LTPR00	FLPR0	WUTMPR0	DMAPR01	DMAPR00	CSIPR001	CSIPR000	CLMPR0

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	LTPR10	FLPR1	WUTMPR1	DMAPR11	DMAPR10	CSIPR101	CSIPR100	CLMPR1

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	PPR08	PPR0LR0	LSPR00	LRPR00

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	PPR18	PPR1LR0	LSPR10	LRPR10

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	TMPR004	CSIPR011 IICPR011	CSIPR010	PPR0LR1	LSPR01	LRPR01	LTPR01	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	TMPR104	CSIPR111 IICPR111	CSIPR110	PPR1LR1	LSPR11	LRPR11	LTPR11	ADPR1

Figure 16-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (78K0R/HG3) (2/3)

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	C0RECPR0	C0WUPPR0[4]	C0ERRPR0[4]	PPR07	PPR06 KRPR0	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	C0RECPR1[4]	C0WUPPR1	C0ERRPR1[4]	PPR17	PPR16 KRPR1	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02H	SRPR02	STPR02 IICPR020	MDPR0	TMPR013	TMPR012	TMPR011	TMPR010	C0TRXPR0

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12H	SRPR12	STPR12 IICPR120	MDPR1	TMPR113	TMPR112	TMPR111	TMPR110	C0TRXPR1[4]

Address: FFFD6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR03L	TMPR022	TMPR021	TMPR020	TMPR017	TMPR016	TMPR015	TMPR014	PPR0R2

Address: FFFD7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR13L	TMPR122	TMPR121	TMPR120	TMPR117	TMPR116	TMPR115	TMPR114	PPR1R2

Address: FFFDEH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR03H	1	1	1	DMAPR03	DMAPR02	TMPR027	TMPR025	TMPR023

Address: FFFDFH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR13H	1	1	1	DMAPR13	DMAPR12	TMPR127	TMPR125	TMPR123

Figure 16-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H(78K0R/HG3)) (3/3)

XXPR1X	XXPR0X	Priority level selection
0	0	Specify level 0 (high priority level)
0	1	Specify level 1
1	0	Specify level 2
1	1	Specify level 3 (low priority level)

Caution The above is the bit layout for the 78K0R/HG3. The available bits differ depending on the product. For details about the bits available for each product, see Table 16-2. Be sure to set bits that are not available to 1.

<R> **(4) External interrupt rising edge enable registers (EGP0, EGP1)**

These registers specify the valid edge for INTP0 to INTP11.

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 16-5. Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1) (1/2)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP0_7	EGP0_6	EGP0_5	EGP0_4	EGP0_3	EGP0_2	EGP0_1	EGP0_0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN0_7	EGN0_6	EGN0_5	EGN0_4	EGN0_3	EGN0_2	EGN0_1	EGN0_0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP1_3	EGP1_2	EGP1_1	EGP1_0

Figure 16-5. Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1) (2/2)

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN1_3	EGN1_2	EGN1_1	EGN1_0

EGPm_n	EGNm_n	INTPn pin valid edge selection (m = 0, 1, n = 0 to 7)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 16-3 shows the ports corresponding to EGPm_n and EGNm_n.

Table 16-3. Ports Corresponding to EGPm_n and EGNm_n

Detection Enable Register		Edge Detection Port		Interrupt Request Signal
EGP0_0	EGN0_0	P120		INTP0
EGP0_1	EGN0_1	P125		INTP1
EGP0_2	EGN0_2	P30 or P31		INTP2
EGP0_3	EGN0_3	P12	78K0R/HC3	INTP3
		P12 or P50	78K0R/HE3, HF3, HG3	
EGP0_4	EGN0_4	P32		INTP4
EGP0_5	EGN0_5	P70		INTP5
EGP0_6	EGN0_6	P71		INTP6
EGP0_7	EGN0_7	P00		INTP7
EGP1_0	EGN1_0	P47	78K0R/HF3, HG3	INTP8[Note]
EGP1_1	EGN1_1	P43	78K0R/HE3, HF3, HG3	INTPR2[Note]
EGP1_2	EGN1_2	P14		INTPLR0
EGP1_3	EGN1_3	P11 or P73		INTPLR1

Note The INTP8 and INTPR2 pins are prohibited be used at the same time.

Caution Select the port mode by clearing EGPm_n and EGNm_n to 0 because an edge may be detected when the external interrupt function is switched to the port function.

Remark m = 0, 1, n = 0 to 7

(5) External interrupt input pin selection register 0 (IPSEL0)

IPSEL0 select the external interrupt input pins from two ports.

IPSEL0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 16-6. Format of External Interrupt Input Pin selection register 0 (IPSEL0)

Address: FFF36H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IPSEL0	0	0	0	0	IPS03[Note]	IPS02	0	0

IPS03[Note]	INTP3 input pin selection
0	P12/INTP3/SO10/TI16/TO16
1	P50/INTP3/TI20/TO20

IPS02	INTP2 input pin selection
0	P30/INTP2/$\overline{\text{SSI00}}$/TI01/TO01
1	P31/INTP2/STOPST/TI11/TO11

Note 78K0R/HE3, 78K0R/HF3, 78K0R/HG3 only. With the 78K0R/HC3 be sure to set to 0.

Caution Rewrite the IPSEL0 register, mask interrupt source associated bit to be rewritten, in order to prevent malfunctioning.

(6) 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 16-7. Configuration of Program Status Word

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	0	After reset
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY	06H

Used when normal instruction is executed

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

16.4 Interrupt Servicing Operations

16.4.1 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 16-4 below.

For the interrupt request acknowledgment timing, see **Figures 16-9** and **16-10**.

Table 16-4. 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.

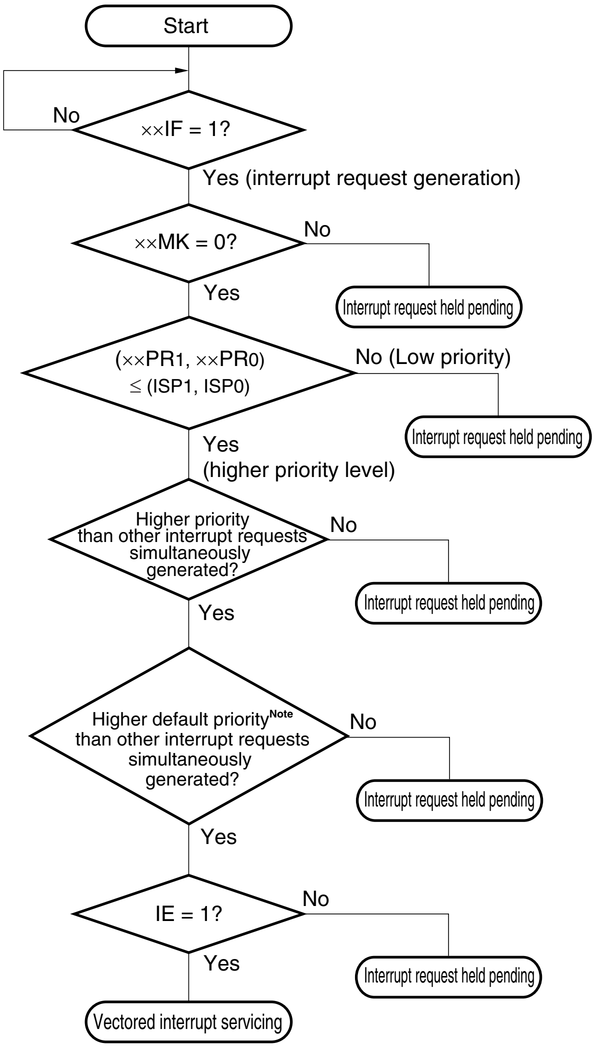
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 16-8 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.

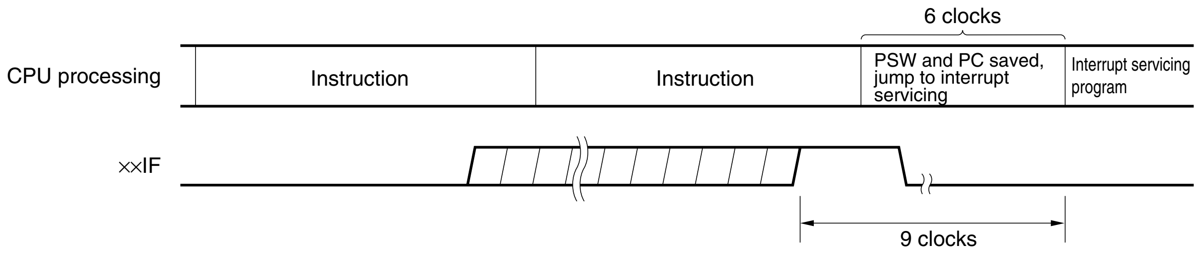
Figure 16-8. Interrupt Request Acknowledgment Processing Algorithm


Start
××IF = 1?
No
Yes (interrupt request generation)
××MK = 0?
No
Yes
Interrupt request held pending
(××PR1, ××PR0)
≤ (ISP1, ISP0)
No (Low priority)
Interrupt request held pending
Yes
(higher priority level)
Higher priority than other interrupt requests simultaneously generated?
No
Interrupt request held pending
Yes
Higher default priority[Note] than other interrupt requests simultaneously generated?
No
Interrupt request held pending
Yes
IE = 1?
No
Yes
Interrupt request held pending
Vectored interrupt servicing

××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 16-7**)

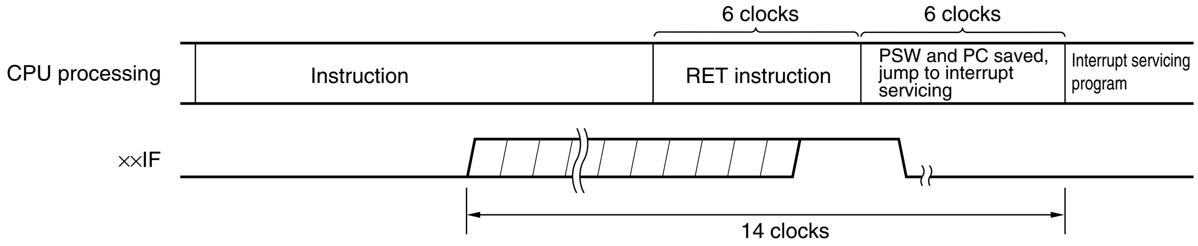
Note For the default priority, see **Table 16-1 Interrupt Source List**.

Figure 16-9. Interrupt Request Acknowledgment Timing (Minimum Time)


6 clocks
CPU processing
Instruction
Instruction
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
××IF
9 clocks

Remark 1 clock: 1/fCLK (fCLK: CPU clock)

Figure 16-10. Interrupt Request Acknowledgment Timing (Maximum Time)


6 clocks
6 clocks
CPU processing
Instruction
RET instruction
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
××IF
14 clocks

Remark 1 clock: 1/fCLK (fCLK: CPU clock)

16.4.2 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.

16.4.3 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 16-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 16-11 shows multiple interrupt servicing examples.

Table 16-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
Interrupt Being Serviced		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.

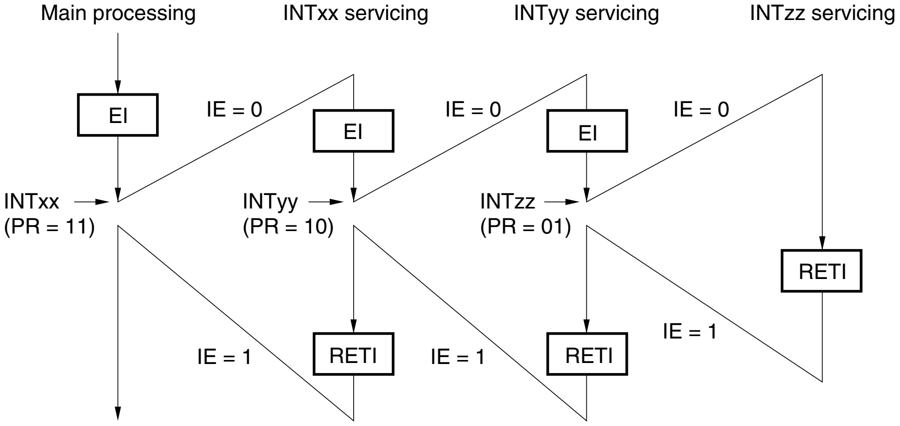
ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.
ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.
ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.
ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment.
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

4. PR is a flag contained in PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

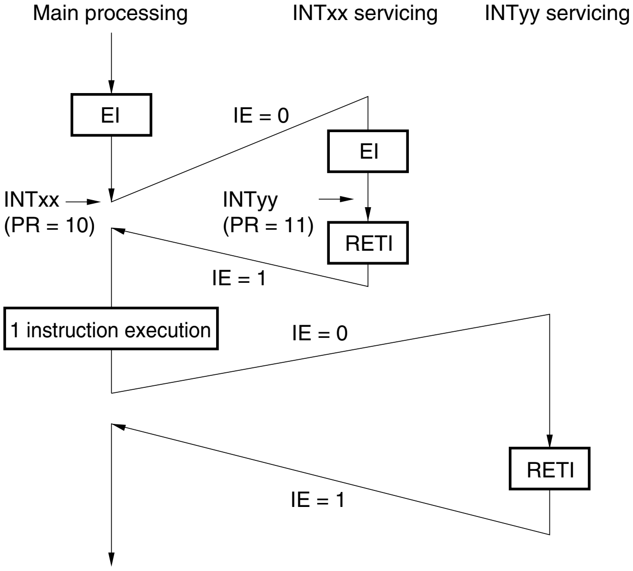
Figure 16-11. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

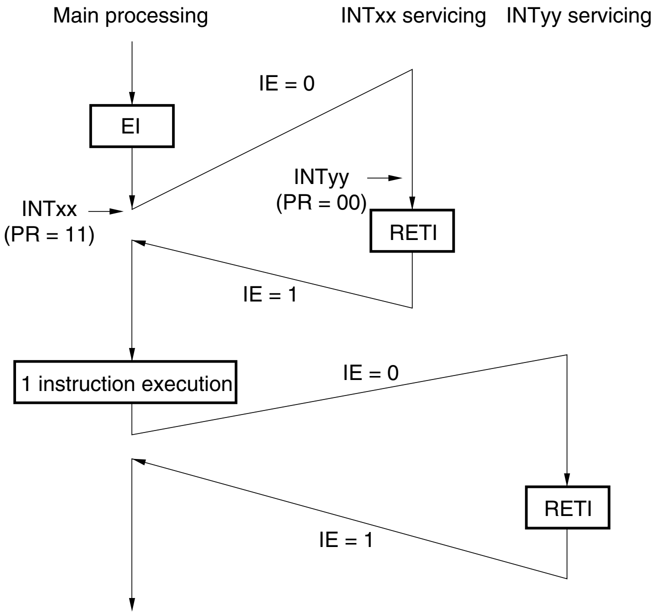


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 16-11. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

16.4.4 Interrupt request hold

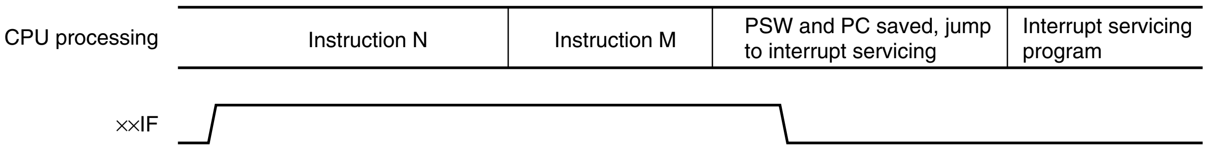
There are instructions where, even if an interrupt request is issued while the instruction are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- Manipulation instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H 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 16-12 shows the timing at which interrupt requests are held pending.

Figure 16-12. Interrupt Request Hold


CPU processing
Instruction N
Instruction M
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
××IF

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 17 KEY INTERRUPT FUNCTION

	78K0R/HC3	78K0R/HE3	78K0R/HF3	78K0R/HG3
Key interrupt	4	8		

17.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by setting the key return mode register (KRM) and inputting a falling edge to the key interrupt input pins (KRn).

Table 17-1. Assignment of Key Interrupt Detection Pins

Flag	Description
KRM_n	Controls KR0 signal in 1-bit units.

Remark n = 0 to 3: 78K0R/HC3
n = 0 to 7: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3

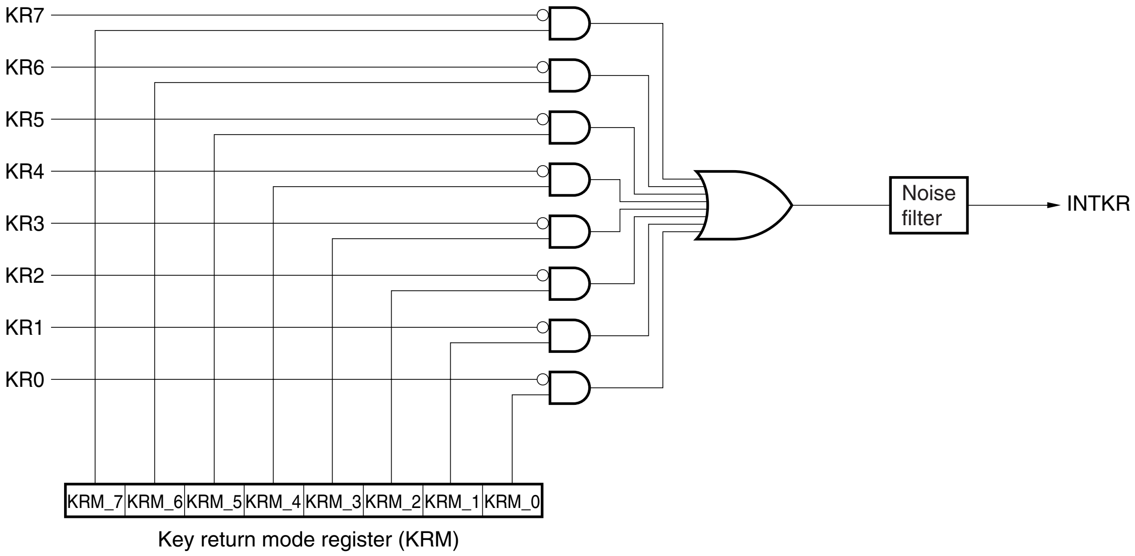
17.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 17-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return mode register (KRM) Port mode register 7 (PM7)

Figure 17-1. Block Diagram of Key Interrupt



Remark KR0 to KR3, KRM_0 to KRM_3: 78K0R/HC3

KR0 to KR7, KRM_0 to KRM_7: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3

17.3 Register Controlling Key Interrupt

<R> **(1) Key return mode register (KRM)**

This register controls the KRMn bits using the KRn signals, respectively.

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

Reset signal generation clears this register to 00H.

Figure 17-2. Format of Key Return Mode Register (KRM)

(1) 78K0R/HC3

Address: FF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	0	0	0	0	KRM_3	KRM_2	KRM_1	KRM_0

(2) 78K0R/HE3, 78K0R/HF3, 78K0R/HG3

Address: FF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	KRM_7	KRM_6	KRM_5	KRM_4	KRM_3	KRM_2	KRM_1	KRM_0

KRM_n	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Cautions 1. If any of the KRM_n bits used is set to 1, set bit n (PU7_n) of the corresponding pull-up resistor register 7 (PU7) to 1 in advance.

2. An interrupt will be generated if the target bit of the KRM register is set while a low level is being input to the key interrupt input pin. To ignore this interrupt, set the KRM register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input low-level width (250 ns or more).

3. The bits not used in the key interrupt mode can be used as normal ports.

Remark n = 0 to 3: 78K0R/HC3

n = 0 to 7: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3

<R> **(2) Port mode register 7 (PM7)**

This register sets the input or output of port 7 in 1-bit units.

When using the P70/INTP5/KR0/TI15/TO15/LVIOUT, P71/INTP6/KR1/TI17/TO17, P72/KR2/CTxD/LTxD1, P73/KR3/CRxD/LRxD1/INTPLR1, P74/KR4/SO01, P75/KR5/SI01, P76/KR6/$\overline{\text{SCK01}}$, P77/KR7/$\overline{\text{SSI01}}$ pins as the key interrupt function, set both PM7_0 to PM7_7 to 1. The output latches of P7_0 to P7_7 at this time may be 0 or 1.

PM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 17-3. Format of Port Mode Register 7 (PM7)

(1) 78K0R/HC3

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	1	1	PM7_3	PM7_2	PM7_1	PM7_0

(2) 78K0R/HE3, 78K0R/HF3, 78K0R/HG3

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM7_7	PM7_6	PM7_5	PM7_4	PM7_3	PM7_2	PM7_1	PM7_0

PM7_n	P7n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution The shaded pins are provided at two ports. Select either port by using the corresponding register.

Remark n = 0 to 3: 78K0R/HC3
n = 0 to 7: 78K0R/HE3, 78K0R/HF3, 78K0R/HG3

CHAPTER 18 STANDBY FUNCTION

18.1 Standby Function and Configuration

18.1.1 Standby function

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, or internal low-speed 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 internal low-speed oscillation clock. The HALT mode can be used when the CPU is operating on either the main system clock or the internal low-speed oscillation 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 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 0 (ADM0) 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 23 OPTION BYTE.

<R> **5. After the STOP mode is released, setting of the PLL is performed by hardware. Therefore setting of the PLL in not required by using software.**

<R> 18.1.2 Registers controlling standby function

One register that performs inverse output of the P31 or P52 port latch when STOP mode is released, and two registers that control the oscillation stabilization time are provided.

- STOP status output control register (STPSTC)
- 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) STOP status output control register (STPSTC)

This register inverts the P31 or P52 port latch when a STOP release source is generated.
STPSTC can be set by a 1-bit or 8-bit memory manipulation instruction.

Caution When using the STOP status output control register, preset the target port to output mode and the port latch to 0.

Figure 18-1. Format of STOP Status Output Control Register (STPSTC)

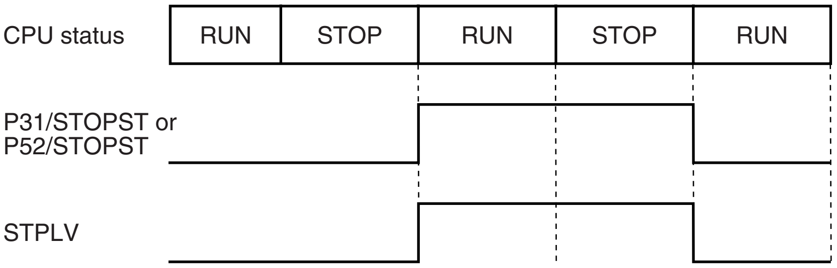
Address: F00FCH After reset: 00H R/W

Symbol	<7>	6	5	<4>	3	2	1	<0>
STPSTC	STPOEN	0	0	STPLV	0	0	0	STPSEL

STPOEN	STPSEL	Operation when STOP released
0	X	Performs no operation when STOP released
1	0	Inverts P31/STOPST pin when STOP release source is generated.
1	1	Inverts P52/STOPST pin when STOP release source is generated.
The STPOEN bit controls output of the STPLV bit value.		

STPLV	Operation when STOP released
0	This logic level is output to the pin set by STPSEL if STPOEN is 1.
1	
The STPLV bit is inverted when a STOP release source is generated, regardless of the STPOEN status. See **Figure 18-2.**	

Figure 18-2. Timings of STPLV, P31/STOPST, and P52/STPST



(2) 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 internal low-speed oscillation are 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.

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

Figure 18-3. 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

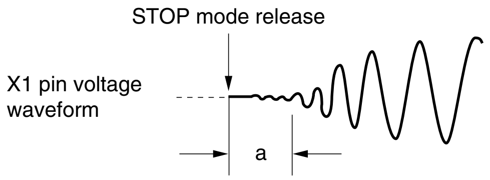
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 8 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^{8}/fx max.	32.0 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^{8}/fx min.	32.0 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^{9}/fx min.	64.0 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	128.0 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	256.0 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	1.02 ms min.	409.6 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	4.10 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	16.38 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	32.77 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. For the following cases, set the OSTS oscillation stabilization time to a value greater than the count value to be checked by using the OSTC register after oscillation starts.

- **When X1 clock oscillation is desired to be started with the internal high-speed oscillation clock or internal low-speed oscillation clock being the CPU clock**
- **When the STOP mode is desired to be released after transitioning to the STOP mode when the internal high-speed oscillation clock is the CPU clock and the X1 clock is also oscillating**
  (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

(3) 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 this register to 07H.

Figure 18-4. 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		
				fx = 8 MHz	fx = 20 MHz
0	0	0	2^8/fx	32.0 μs	Setting prohibited
0	0	1	2^9/fx	64.0 μs	25.6 μs
0	1	0	2^{10}/fx	128.0 μs	51.2 μs
0	1	1	2^{11}/fx	256.0 μs	102.4 μs
1	0	0	2^{13}/fx	1.02 ms	409.6 μs
1	0	1	2^{15}/fx	4.10 ms	1.64 ms
1	1	0	2^{17}/fx	16.38 ms	6.55 ms
1	1	1	2^{18}/fx	32.77 ms	13.11 ms

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

2. Use the OSTS register to set a suitable oscillation stabilization time before oscillating the X1 clock by setting the MSTOP bit.

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

4. Before changing the setting of the OSTS register, confirm that the count operation of the OSTC register is completed.

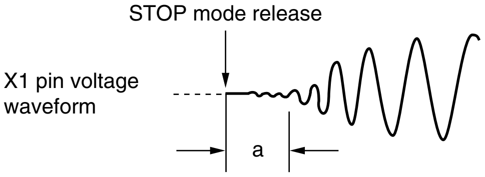
5. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

6. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. For the following cases, set the OSTS oscillation stabilization time to a value greater than the count value to be checked by using the OSTC register after oscillation starts.

- **When X1 clock oscillation is desired to be started with the internal high-speed oscillation clock or internal low-speed oscillation clock being the CPU clock**
- **When the STOP mode is desired to be released after transitioning to the STOP mode when the internal high-speed oscillation clock is the CPU clock and the X1 clock is also oscillating**
  (Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.)

7. If the STOP mode is released when the PLL operates, a time of the oscillation stabilization time set by using OSTS plus the PLL lockup wait time is required.

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



Remark fx: X1 clock oscillation frequency

18.2 Standby Function Operation

18.2.1 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, or internal low-speed oscillation clock.

The operating statuses in the HALT mode are shown below.

Table 18-1. Operating Statuses in HALT Mode (1/2)

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})	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}	Status before HALT mode was set is retained		
		f_X			
		f_{EX}			
	Subclock	f_{EXS}			
	f_{IL}				
	f_{PLL}				
CPU			Operation stopped		
Code flash memory			Operable in low-current consumption mode		
RAM			Operation stopped. However, 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		
16-bit wakeup timer			Operable (depends on the operation clock status)		
Timer array unit (TAU)			Operable		
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) and bits 6 (LIOSTOPB) and 7 (LIOUSE) of option byte (000C1H) See **5.7.5 Internal low-speed oscillation clock** or **8.4.1 Controlling operation of watchdog timer** for the operating status.		
Clock output			Operable		
A/D converter					
Serial array unit (SAU)					
LIN-UART					
CAN controller					
Multiplier/divider					
DMA controller					
Power-on-clear function					
Low-voltage detection function					
External interrupt					
Key interrupt function					
Illegal-memory access detection function			Operable (detection of illegal access by DMA possible during HALT mode)		

Remarks 1. f_{IH}: Internal high-speed oscillation clock

f_X: X1 clock

f_{EX}: External main system clock

f_{EXS}: External Subclock

f_{IL}: Internal low-speed oscillation clock

f_{PLL}: PLL clock

2. The functions mounted depend on the product. See **1.7 Block Diagram** and **1.8 Outline of Functions**.

Table 18-1. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting Item			When HALT Instruction Is Executed While CPU Is Operating on Internal Low-speed Oscillation Clock
			When CPU Is Operating on Internal Low-speed Oscillation Clock (f_{IL})
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}	
	Subclock	f_{EXS}	
	f_{IL}		
	f_{PLL}		
CPU			Operation stopped
Code flash memory			Operable in low-current consumption mode
RAM			Operation stopped. However, 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
16-bit wakeup timer			Operable (depends on the operation clock status)
Timer array unit (TAU)			Operable
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) and bits 6 (LIOSTOPB) and 7 (LIOUSE) of option byte (000C1H) See **5.7.5 Internal low-speed oscillation clock** or **8.4.1 Controlling operation of watchdog timer** for the operating status.
Clock output			Operable
A/D converter			
Serial array unit (SAU)			
LIN-UART			
CAN controller			
Multiplier/divider			
DMA controller			
Power-on-clear function			
Low-voltage detection function			
External interrupt			
Key interrupt function			
Illegal-memory access detection function			Operable (detection of illegal access by DMA possible during HALT mode)

Remarks 1. f_{IH}: Internal high-speed oscillation clock
f_{X}: X1 clock
f_{EX}: External main system clock
f_{EXS}: External Subclock
f_{IL}: Internal low-speed oscillation clock
f_{PLL}: PLL clock

2. The functions mounted depend on the product. See **1.7 Block Diagram** and **1.8 Outline of Functions**.

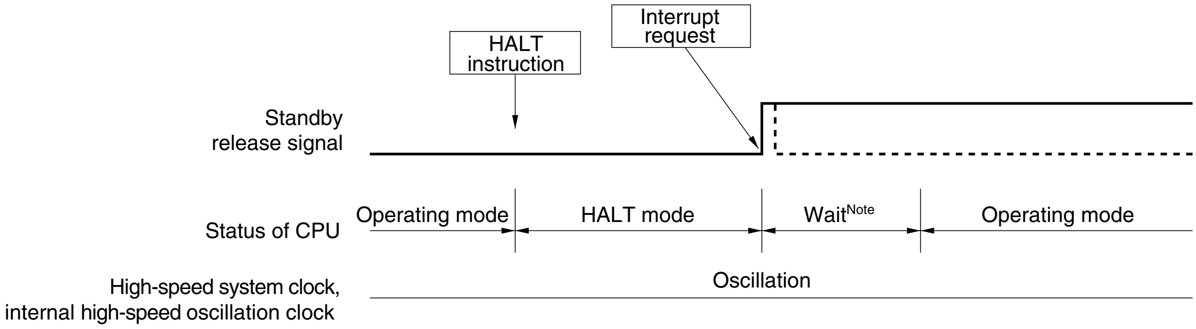
(2) HALT mode release

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 18-5. HALT Mode Release by Interrupt Request Generation



Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 10 to 12 clocks
- When vectored interrupt servicing is not carried out: 5 or 6 clocks

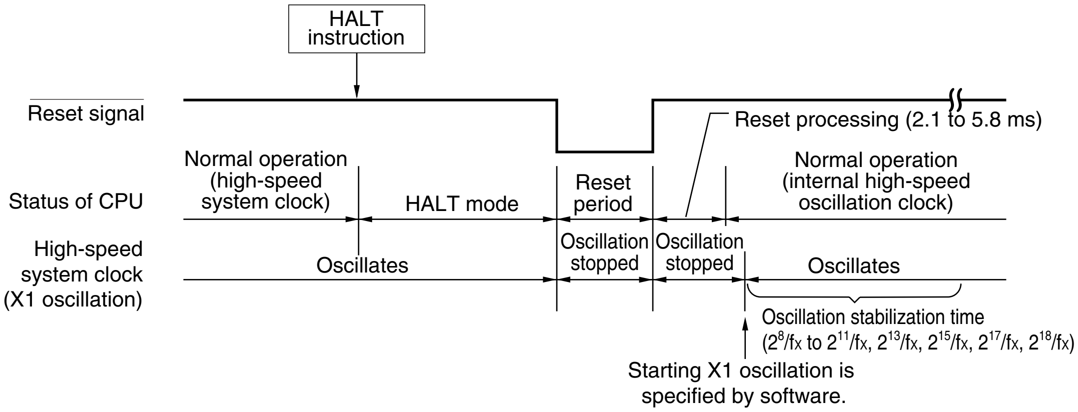
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

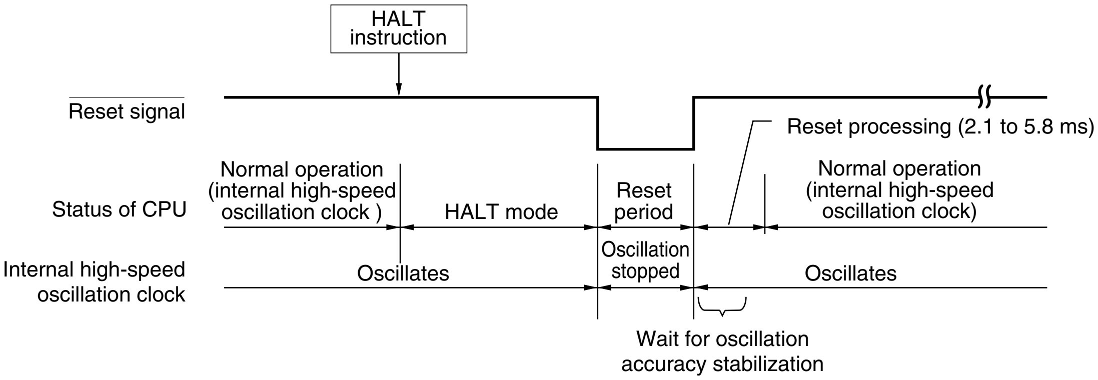
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 18-6. HALT Mode Release by Reset

(1) When high-speed system clock is used as CPU clock


HALT instruction
Reset signal
Reset processing (2.1 to 5.8 ms)
Status of CPU
Normal operation (high-speed system clock)
HALT mode
Reset period
Normal operation (internal high-speed oscillation clock)
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time
(2^8/fx to 2^11/fx, 2^13/fx, 2^15/fx, 2^17/fx, 2^18/fx)
Starting X1 oscillation is specified by software.

(2) When internal high-speed oscillation clock or double-speed mode internal high-speed oscillation clock is used as CPU clock


HALT instruction
Reset signal
Reset processing (2.1 to 5.8 ms)
Status of CPU
Normal operation (internal high-speed oscillation clock)
HALT mode
Reset period
Normal operation (internal high-speed oscillation clock)
Internal high-speed oscillation clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

Remark fx: X1 clock oscillation frequency

18.2.2 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.

Caution 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.

The operating statuses in the STOP mode are shown below.

Table 18-2. Operating Statuses in STOP Mode

HALT Mode Setting Item			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}			
	Subclock	f_{EXS}	Status before STOP mode was set is retained		
	f_{IL}		Set by bits 6 (LIOSTOP) and 7 (LIOUSE) of option byte (000C1H) • LIOSTOP = 0 or LIOUSE = 0: Stops • LIOSTOP = 1 and LIOUSE = 1: Operates		
	f_{PLL}		Stopped		
CPU			Operation stopped		
Code flash memory			Operation stopped		
RAM			Operation stopped. However, 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		
16-bit wakeup timer			Operable (depends on the operation clock status)		
Timer array unit (TAU)			Operable		
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) and bits 6 (LIOSTOPB) and 7 (LIOUSE) of option byte (000C1H) See **5.7.5 Internal low-speed oscillation clock** or **8.4.1 Controlling operation of watchdog timer** for the operating status.		
Clock output			Operable (depends on the operation clock status)		
A/D converter			Operation stopped		
Serial array unit (SAU)					
LIN-UART					
CAN controller					
Multiplier/divider					
DMA controller					
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					
Key interrupt function					
Illegal-memory access detection function			Operation stopped		

Remarks 1. f_{IH}: Internal high-speed oscillation clock

f_X: X1 clock

f_{EX}: External main system clock

f_{EXS}: External Subclock

f_{IL}: Internal low-speed oscillation clock

f_{PLL}: PLL clock

2. The functions mounted depend on the product. See **1.7 Block Diagram** and **1.8 Outline of Functions**.

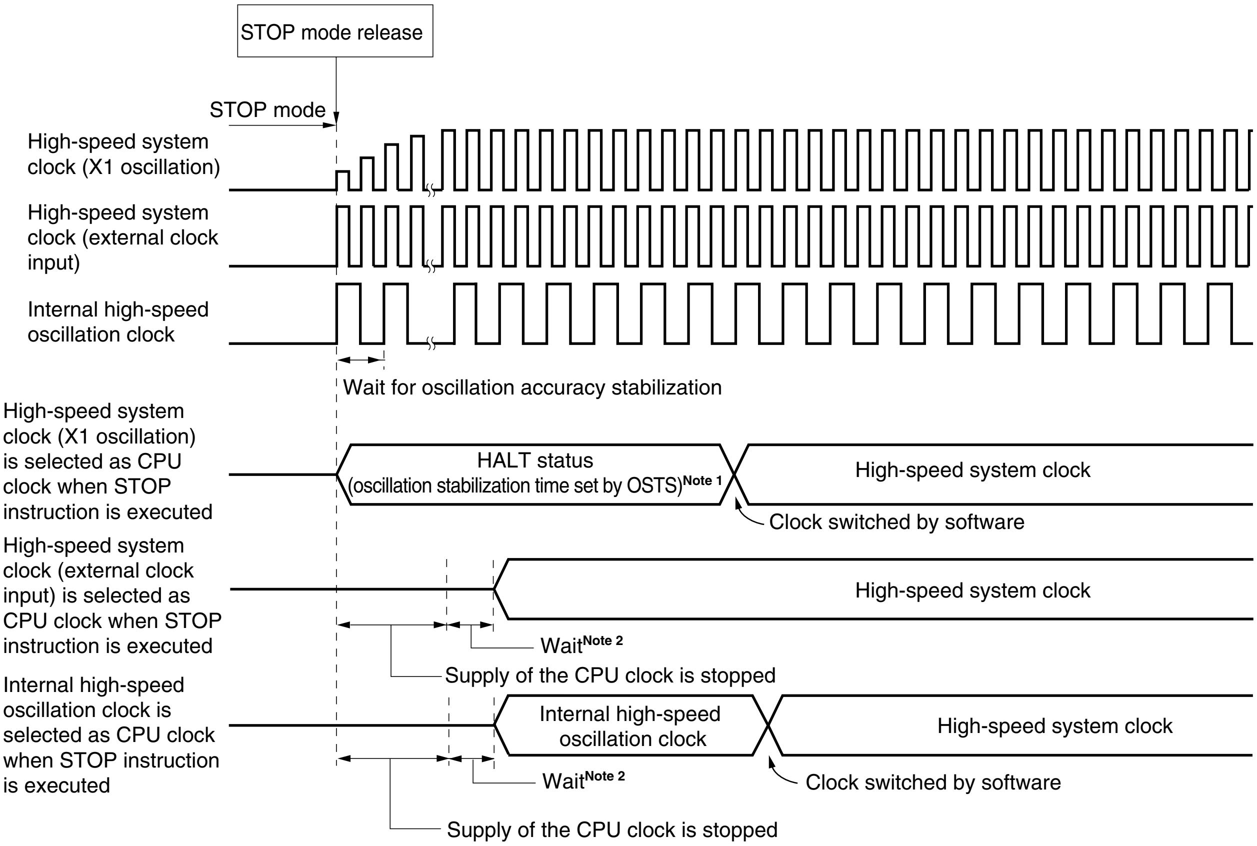
Cautions 1. To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.

2. To stop the internal low-speed oscillation clock in the STOP mode, set an option byte (bit 6 (LIOSTOPB) of 000C1H = 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).

(2) STOP mode release

Figure 18-7. Operation Timing When STOP Mode Is Released


STOP mode release
STOP mode
High-speed system clock (X1 oscillation)
High-speed system clock (external clock input)
Internal high-speed oscillation clock
Wait for oscillation accuracy stabilization
High-speed system clock (X1 oscillation) is selected as CPU clock when STOP instruction is executed
HALT status (oscillation stabilization time set by OSTS)Note 1
High-speed system clock
Clock switched by software
High-speed system clock (external clock input) is selected as CPU clock when STOP instruction is executed
High-speed system clock
WaitNote 2
Supply of the CPU clock is stopped
Internal high-speed oscillation clock is selected as CPU clock when STOP instruction is executed
Internal high-speed oscillation clock
High-speed system clock
WaitNote 2
Clock switched by software
Supply of the CPU clock is stopped

Notes 1. 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.”

2. The wait time is as follows:

- When vectored interrupt servicing is carried out : 10 to 12 clocks
- When vectored interrupt servicing is not carried out : 5 or 6 clocks

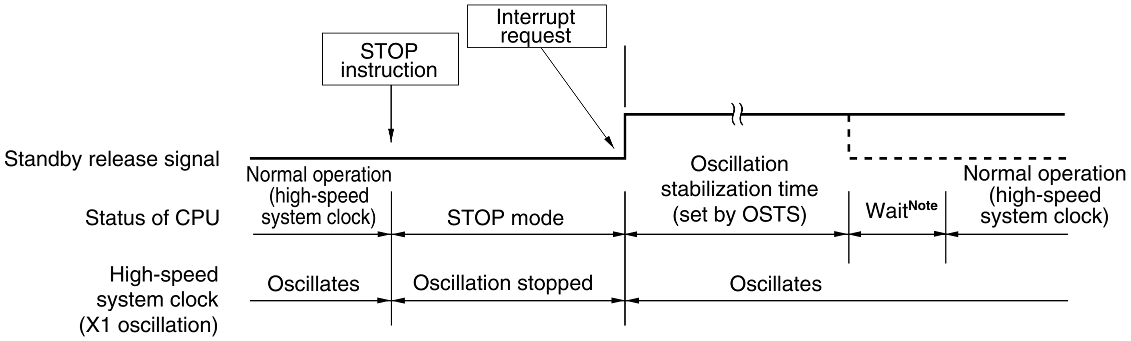
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

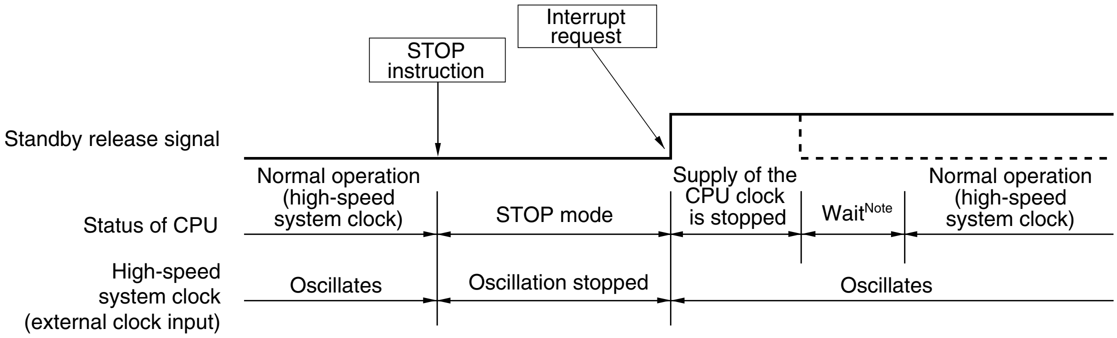
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 18-8. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed system clock (X1 oscillation) is used as CPU clock


STOP instruction
Interrupt request
Standby release signal
Status of CPU
High-speed system clock (X1 oscillation)
Normal operation (high-speed system clock)
STOP mode
Oscillation stabilization time (set by OSTS)
Wait^Note
Normal operation (high-speed system clock)
Oscillates
Oscillation stopped
Oscillates

(2) When high-speed system clock (external clock input) is used as CPU clock


STOP instruction
Interrupt request
Standby release signal
Status of CPU
High-speed system clock (external clock input)
Normal operation (high-speed system clock)
STOP mode
Supply of the CPU clock is stopped
Wait^Note
Normal operation (high-speed system clock)
Oscillates
Oscillation stopped
Oscillates

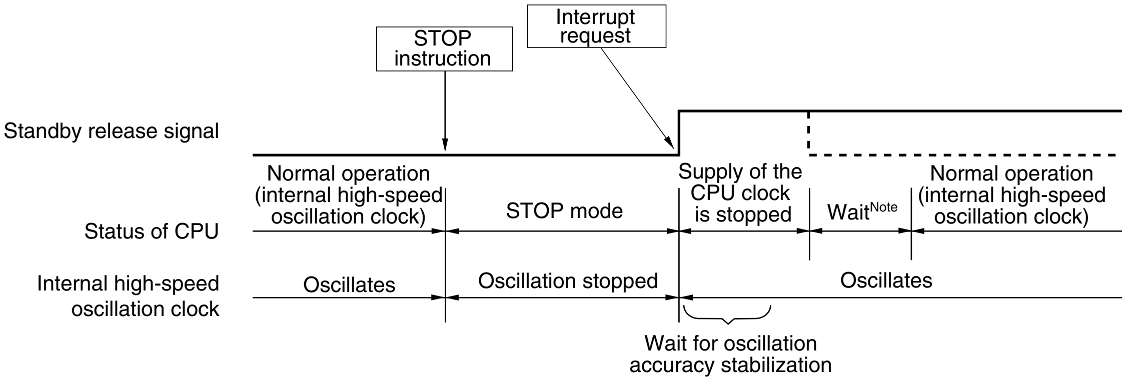
Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 10 to 12 clocks
- When vectored interrupt servicing is not carried out: 5 or 6 clocks

Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 18-8. STOP Mode Release by Interrupt Request Generation (2/2)

(3) When internal high-speed oscillation clock is used as CPU clock


STOP instruction
Interrupt request
Standby release signal
Normal operation (internal high-speed oscillation clock)
STOP mode
Supply of the CPU clock is stopped
Wait[Note]
Normal operation (internal high-speed oscillation clock)
Status of CPU
Internal high-speed oscillation clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 10 to 12 clocks
- When vectored interrupt servicing is not carried out: 5 or 6 clocks

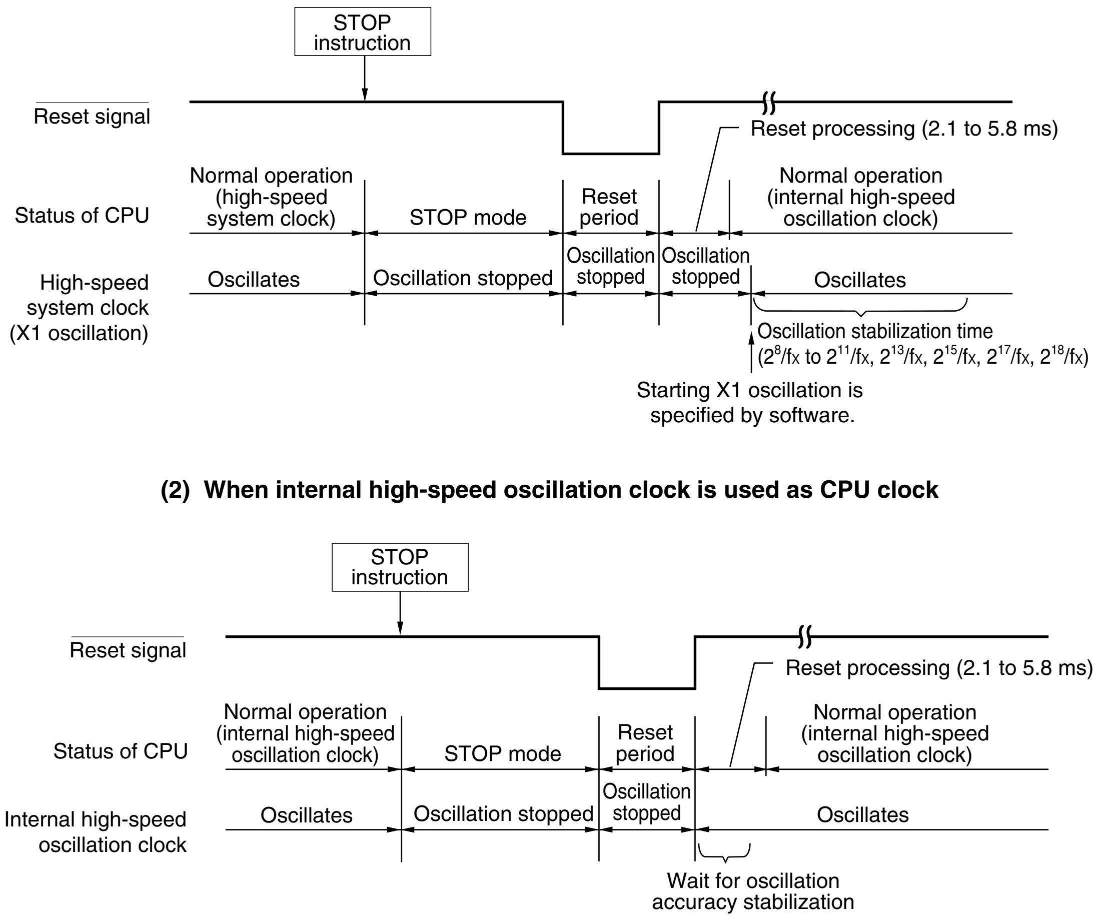
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(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 18-9. STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock


STOP instruction
Reset signal
Reset processing (2.1 to 5.8 ms)
Status of CPU
Normal operation (high-speed system clock)
STOP mode
Reset period
Normal operation (internal high-speed oscillation clock)
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time (2^8/fx to 2^{11}/fx, 2^{13}/fx, 2^{15}/fx, 2^{17}/fx, 2^{18}/fx)
Starting X1 oscillation is specified by software.
STOP instruction
Reset signal
Reset processing (2.1 to 5.8 ms)
Status of CPU
Normal operation (internal high-speed oscillation clock)
STOP mode
Reset period
Normal operation (internal high-speed oscillation clock)
Internal high-speed oscillation clock
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

(2) When internal high-speed oscillation clock is used as CPU clock

Remark fx: X1 clock oscillation frequency

CHAPTER 19 RESET FUNCTION

The following seven operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-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]
(6) Internal reset by detection of main clock oscillation stop via clock monitoring
(7) Internal reset by detection of illegal memory access

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, by POC and LVI circuit voltage detection, execution of illegal instruction[Note], by detection of a main clock oscillation stop via clock monitoring, or by detection of an illegal memory access, and each item of hardware is set to the status shown in Tables 19-1 and 19-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release, except for P130 and P140, which is low-level output.

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 19-2** to **19-4**) after reset processing. Reset by POC and LVI circuit supply voltage detection is automatically released when $V_{DD} \geq V_{POC}$ or $V_{DD} \geq V_{LVI}$ after the reset, and program execution starts using the internal high-speed oscillation clock (see **CHAPTER 20 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 21 LOW-VOLTAGE DETECTOR**) after reset processing.

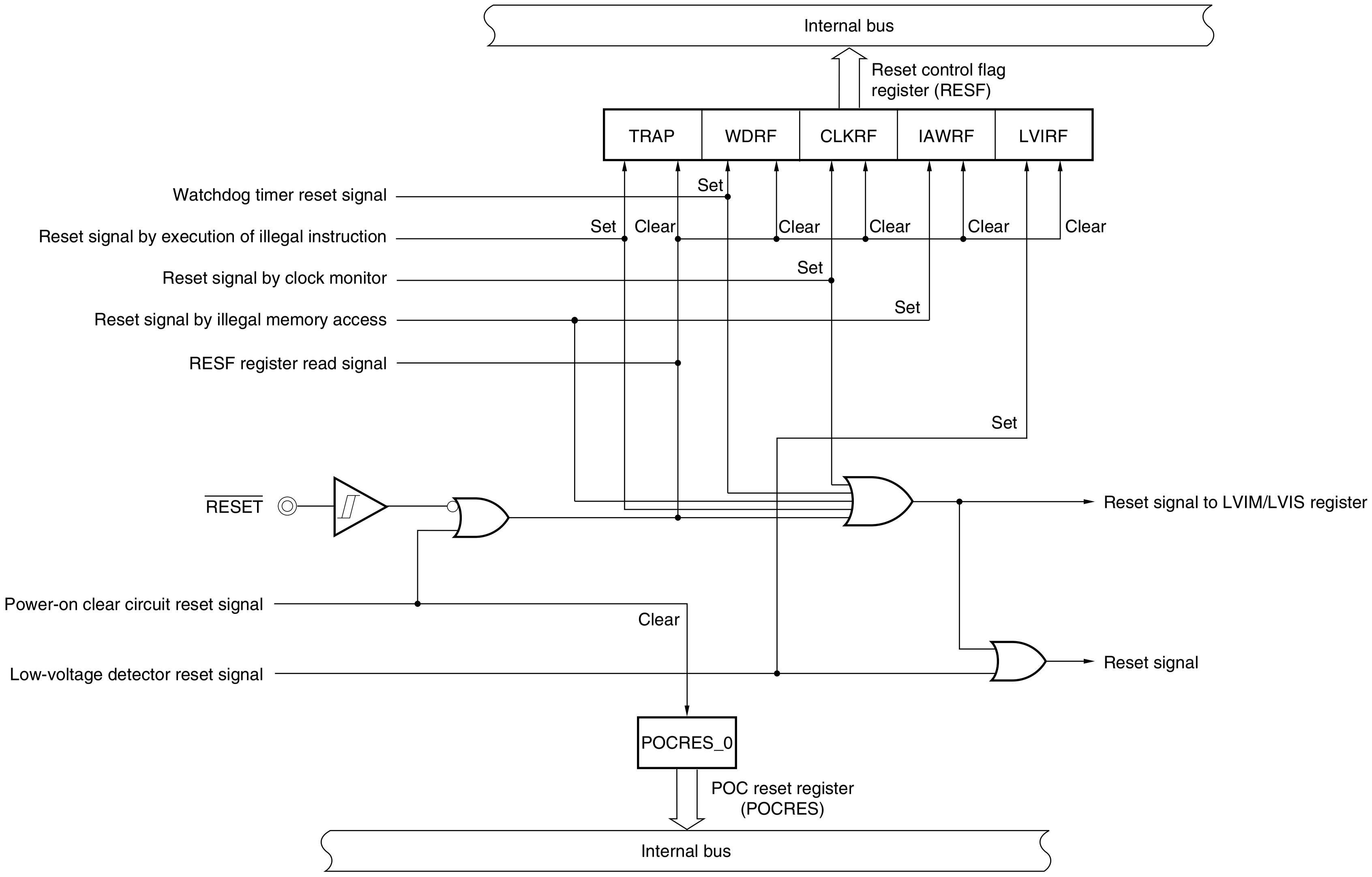
Note The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
(If an external reset is effected upon power application, the period during which the supply voltage is outside the operating range (V_{DD} < 2.7 V) is not counted in the 10 μs. However, the low-level input may be continued before POC is released.)

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. However, because SFR and 2nd SFR are initialized, the port pins become high-impedance, except for P130 and P140, which is set to low-level output.

Figure 19-1. Block Diagram of Reset Function

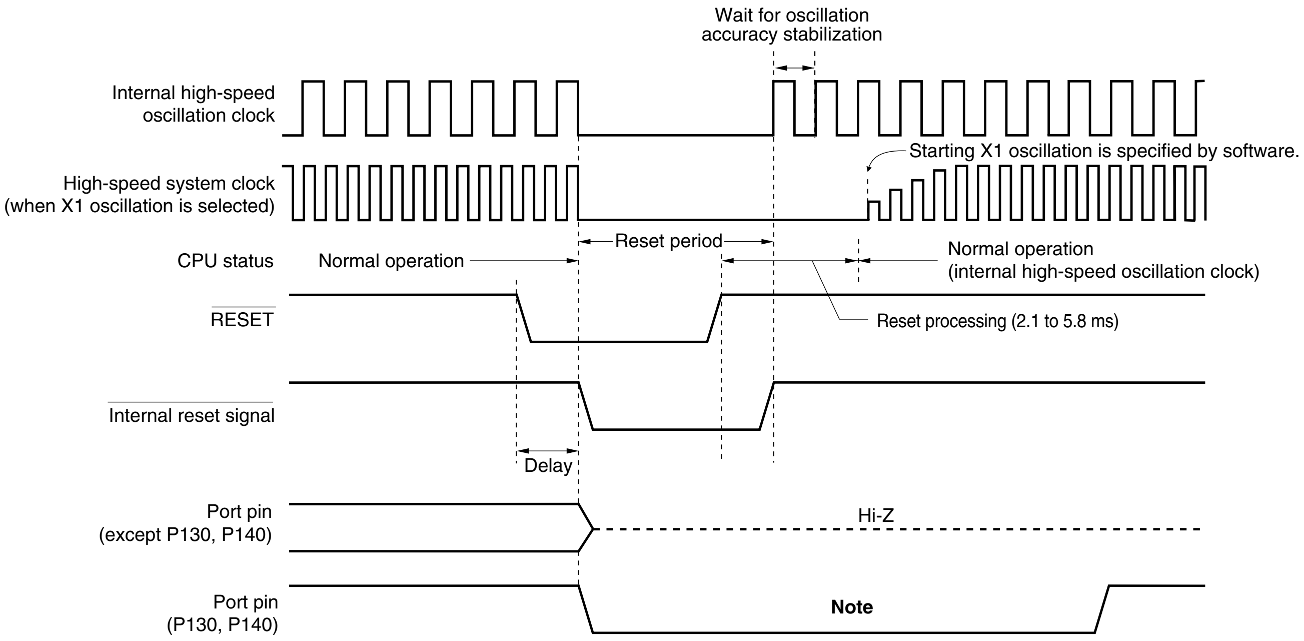

Internal bus
Reset control flag
register (RESF)
TRAP
WDRF
CLKRF
IAWRF
LVIRF
Set
Set
Clear
Clear
Clear
Clear
Clear
Set
Set
Set
Watchdog timer reset signal
Reset signal by execution of illegal instruction
Reset signal by clock monitor
Reset signal by illegal memory access
RESF register read signal
RESET
Reset signal to LVIM/LVIS register
Power-on clear circuit reset signal
Clear
Low-voltage detector reset signal
Reset signal
POCRES_0
POC reset register
(POCRES)
Internal bus

Caution An LVI circuit internal reset does not reset the LVI circuit.

Remarks 1. LVIM: Low-voltage detection register

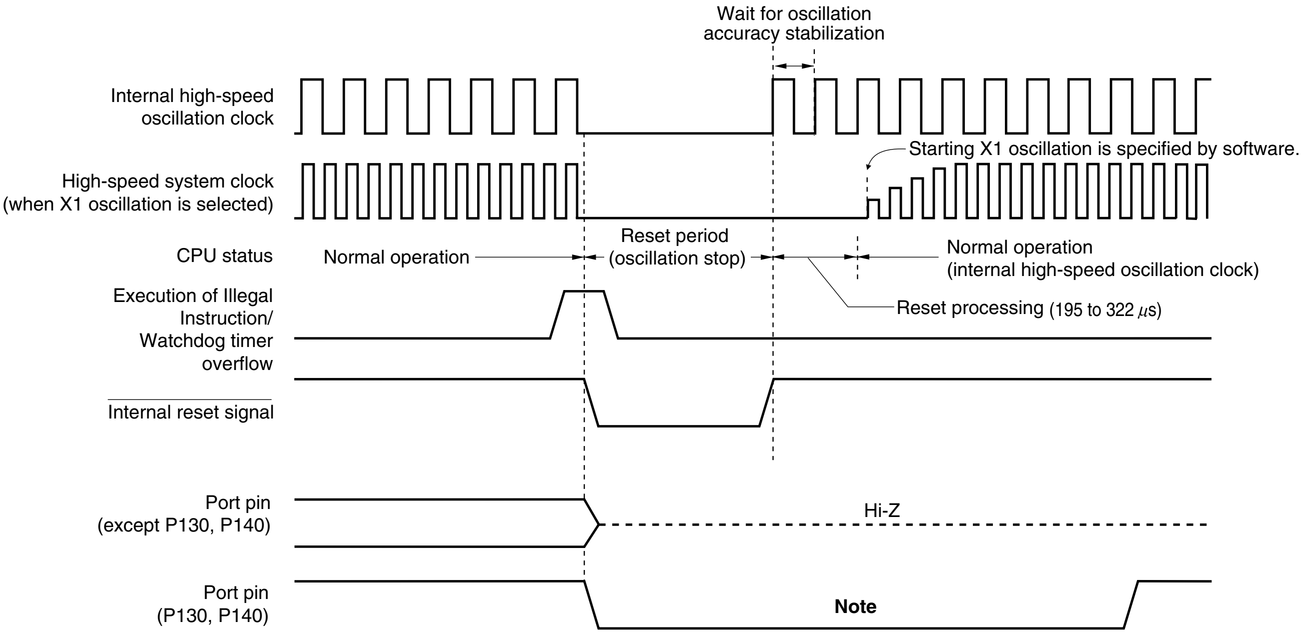
2. LVIS: Low-voltage detection level select register

Figure 19-2. Timing of Reset by $\overline{\text{RESET}}$ Input


Wait for oscillation accuracy stabilization
Internal high-speed oscillation clock
High-speed system clock (when X1 oscillation is selected)
Starting X1 oscillation is specified by software.
CPU status
Normal operation
Reset period
Normal operation (internal high-speed oscillation clock)
RESET
Reset processing (2.1 to 5.8 ms)
Internal reset signal
Delay
Port pin (except P130, P140)
Hi-Z
Port pin (P130, P140)
Note

Note When P130 and P140 are set to high-level output before reset is effected, the output signals of P130 and P140 can be dummy-output as a reset signal to an external device, because P130 and P140 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 and P140 to high-level output by software.

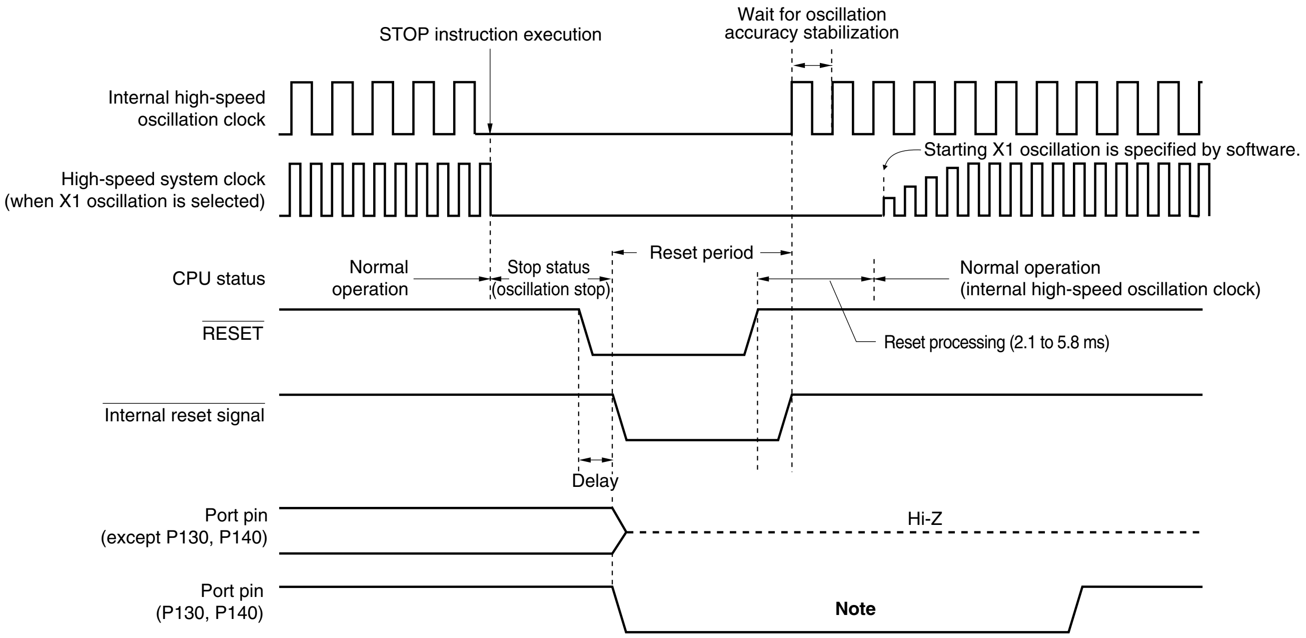
Figure 19-3. Timing of Reset Due to Execution of Illegal Instruction or Watchdog Timer Overflow



Note When P130 and P140 are set to high-level output before reset is effected, the output signals of P130 and P140 can be dummy-output as a reset signal to an external device, because P130 and P140 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 and P140 to high-level output by software.

Caution A watchdog timer internal reset resets the watchdog timer.

Figure 19-4. Timing of Reset in STOP Mode by RESET Input



Note When P130 and P140 are set to high-level output before reset is effected, the output signals of P130 and P140 can be dummy-output as a reset signal to an external device, because P130 and P140 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 and P140 to high-level output by software.

Remark For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 20 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 21 LOW-VOLTAGE DETECTOR**.

Table 19-1. 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)
	Subclock	f_{EXS}	Operation stopped
	f_{IL}		
	f_{PLL}		
CPU			
Code flash memory			Operable in low-current consumption mode
RAM			Operation stopped (however, retention at the POC detection voltage or more is performed)
Port (latch)			Operation stopped (ports other than P130 and P140 are Hi-Z (input disabled). P130 and P140 output a low level.)
16-bit wakeup timer			Operation stopped
Timer array unit (TAU)			
Watchdog timer			
Clock output			
A/D converter			
Serial array unit (SAU)			
LIN-UART			
CAN controller			
Multiplier/divider			
DMA controller			
Power-on-clear function			Operable
Low-voltage detection function			Operation stopped (however, when the LVI default operation is valid and operation continues at LVI reset)
External interrupt			Operation stopped
Key interrupt function			
Illegal-memory access detection function			Operation stopped

Remarks 1. f_{IH}: Internal high-speed oscillation clock

f_X: X1 clock

f_{EX}: External main system clock

f_{EXS}: External Subclock

f_{IL}: Internal low-speed oscillation clock

f_{PLL}: PLL clock

2. The functions mounted depend on the product. See **1.7 Block Diagram** and **1.8 Outline of Functions**.

Table 19-2. 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, P1, P3 to P10, P12 to P15) (output latches)		00H
Port mode registers (PM0, PM1, PM3 to PM10, PM12, PM15)		FFH
Port mode register (PM14)		FEH
Port input mode registers 6, 7 (PIM6, PIM7)		00H
Port output mode registers 4, 7 (POM4, POM7)		00H
Pull-up resistor option registers (PU0, PU1, PU3 to PU7, PU12, PU14, PU15)		00H
Clock operation mode control register (CMC)		00H
Clock operation status control register (CSC)		C0H
Processor mode control register (PMC)		00H
System clock control register (CKC)		01H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		07H
Noise filter enable registers 0, 1, 2, 3 (NFEN0, NFEN1, NFEN2, NFEN3)		00H
Peripheral enable registers 0, 1 (PER0, PER1)		00H
Peripheral clock select register (PCKSEL)		00H
PLL control register (PLLCTL)		00H
PLL status register (PLLSTS)		00H
Operation speed mode control register (OSMC)		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. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

Remark The special function register (SFR) mounted depend on the product. See **3.2.4 Special function registers (SFRs)** and **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 19-2. Hardware Statuses After Reset Acknowledgment (2/6)

Hardware		After Reset Acknowledgment[Note 1]
Timer array unit (TAU)	Timer data registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27 (TDR00, TDR01, TDR02, TDR03, TDR04, TDR05, TDR06, TDR07, TDR10, TDR11, TDR12, TDR13, TDR14, TDR15, TDR16, TDR17, TDR20, TDR21, TDR22, TDR23, TDR24, TDR25, TDR26, TDR27)	0000H
	Timer mode registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27 (TMR00, TMR01, TMR02, TMR03, TMR04, TMR05, TMR06, TMR07, TMR10, TMR11, TMR12, TMR13, TMR14, TMR15, TMR16, TMR17, TMR20, TMR21, TMR22, TMR23, TMR24, TMR25, TMR26, TMR27)	0000H
	Timer status registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27 (TSR00, TSR01, TSR02, TSR03, TSR04, TSR05, TSR06, TSR07, TSR10, TSR11, TSR12, TSR13, TSR14, TSR15, TSR16, TSR17, TSR20, TSR21, TSR22, TSR23, TSR24, TSR25, TSR26, TSR27)	0000H
	Timer input select registers 0, 1 (TIS0, TIS1)	00H
	Timer counter registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27 (TCR00, TCR01, TCR02, TCR03, TCR04, TCR05, TCR06, TCR07, TCR10, TCR11, TCR12, TCR13, TCR14, TCR15, TCR16, TCR17, TCR20, TCR21, TCR22, TCR23, TCR24, TCR25, TCR26, TCR27)	FFFFH
	Timer channel enable status registers 0, 1, 2 (TE0, TE1, TE2)	0000H
	Timer channel start registers 0, 1, 2 (TS0, TS1, TS2)	0000H
	Timer channel stop registers 0, 1, 2 (TT0, TT1, TT2)	0000H
	Timer clock select registers 0, 1, 2 (TPS0, TPS1, TPS2)	0000H
	Timer output registers 0, 1, 2 (TO0, TO1, TO2)	0000H
	Timer output enable registers 0, 1, 2 (TOE0, TOE1, TOE2)	0000H
	Timer output level registers 0, 1, 2 (TOL0, TOL1, TOL2)	0000H
	Timer output mode registers 0, 1, 2 (TOM0, TOM1, TOM2)	0000H
16-bit wakeup timer	WUTM control register (WUTMCTL)	00H
	WUTM compare register (WUTMCMP)	0000H
Clock output controller	Clock output select register (CKS)	00H [Note 2]
Watchdog timer	Enable register (WDTE)	1AH/9AH [Note 3]

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 CKS varies depending on the reset source.

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

Remark The special function register (SFR) mounted depend on the product. See **3.2.4 Special function registers (SFRs)** and **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 19-2. Hardware Statuses After Reset Acknowledgment (3/6)

Hardware		Status After Reset Acknowledgment[Note 1]
A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	10-bit A/D conversion result registers 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 (ADCR0, ADCR1, ADCR2, ADCR3, ADCR4, ADCR5, ADCR6, ADCR7, ADCR8, ADCR9, ADCR10, ADCR11, ADCR12, ADCR13, ADCR14, ADCR15, ADCR16, ADCR17, ADCR18, ADCR19, ADCR20, ADCR21, ADCR22, ADCR23)	0000H
	8-bit A/D conversion result registers 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 (ADCR0H, ADCR1H, ADCR2H, ADCR3H, ADCR4H, ADCR5H, ADCR6H, ADCR7H, ADCR8H, ADCR9H, ADCR10H, ADCR11H, ADCR12H, ADCR13H, ADCR14H, ADCR15H, ADCR16H, ADCR17H, ADCR18H, ADCR19H, ADCR20H, ADCR21H, ADCR22H, ADCR23H)	00H
	A/D converter mode register 0 (ADM0)	00H
	A/D converter mode register 1 (ADM1)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	00H[Note 2]
	A/D conversion time setting register (ADSMP)	00H
Serial array unit (SAU)	Serial data registers 00, 01, 10, 11, 20, 21 (SDR00, SDR01, SDR10, SDR11, SDR20, SDR21)	0000H
	Serial status registers 00, 01, 10, 11, 20, 21 (SSR00, SSR01, SSR10, SSR11, SSR20, SSR21)	0000H
	Serial flag clear trigger registers 00, 01, 10, 11, 20, 21 (SIR00, SIR01, SIR10, SIR11, SIR20, SIR21)	0000H
	Serial mode registers 00, 01, 10, 11, 20, 21 (SMR00, SMR01, SMR10, SMR11, SMR20, SMR21)	0020H
	Serial communication operation setting registers 00, 01, 10, 11, 20, 21 (SCR00, SCR01, SCR10, SCR11, SCR20, SCR21)	0087H
	Serial channel enable status registers 0, 1, 2 (SE0, SE1, SE2)	0000H
	Serial channel start registers 0, 1, 2 (SS0, SS1, SS2)	0000H
	Serial channel stop registers 0, 1, 2 (ST0, ST1, ST2)	0000H
	Serial clock select registers 0, 1, 2 (SPS0, SPS1, SPS2)	0000H
	Serial output registers 0, 1, 2 (SO0, SO1, SO2)	0303H
	Serial output enable registers 0, 1, 2 (SOE0, SOE1, SOE2)	0000H
	Serial output level registers 0, 1, 2 (SOL0, SOL1, SOL2)	0000H
	Serial slave select enable register 0 (SSE0)	0000H
	Serial communication pin select register (STSEL)	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 ADPC register is not reset even if PER0.ADCEN = 0 is set.

Remark The special function register (SFR) mounted depend on the product. See **3.2.4 Special function registers (SFRs)** and **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 19-2. Hardware Statuses After Reset Acknowledgment (4/6)

Hardware		Status After Reset Acknowledgment[Note]
LIN-UART0	LIN-UART0 control register 0 (UF0CTL0)	10H
	LIN-UART0 control register 1 (UF0CTL1)	0FFFH
	LIN-UART0 option register 0 (UF0OPT0)	14H
	LIN-UART0 option register 1 (UF0OPT1)	00H
	LIN-UART0 option register 2 (UF0OPT2)	00H
	LIN-UART0 status register (UF0STR)	0000H
	LIN-UART0 status clear register (UF0STC)	0000H
	LIN-UART0 receive data register (UF0RX)	0000H
	LIN-UART0 8-bit receive data register (UF0RXB)	00H
	LIN-UART0 transmit data register (UF0TX)	0000H
	LIN-UART0 8-bit transmit data register (UF0TXB)	00H
	LIN-UART0 wait transmit data register (UF0WTX)	0000H
	LIN-UART0 8-bit wait transmit data register (UF0WTXB)	00H
	LIN-UART0 ID setting register (UF0ID)	00H
	LIN-UART0 buffer registers 0, 1, 2, 3, 4, 5, 6, 7, 8 (UF0BUF0, UF0BUF1, UF0BUF2, UF0BUF3, UF0BUF4, UF0BUF5, UF0BUF6, UF0BUF7, UF0BUF8)	00H
	LIN-UART0 buffer control register (UF0BUCTL)	0000H
LIN-UART1	LIN-UART1 control register 0 (UF1CTL0)	10H
	LIN-UART1 control register 1 (UF1CTL1)	0FFFH
	LIN-UART1 option register 0 (UF1OPT0)	14H
	LIN-UART1 option register 1 (UF1OPT1)	00H
	LIN-UART1 option register 2 (UF1OPT2)	00H
	LIN-UART1 status register (UF1STR)	0000H
	LIN-UART1 status clear register (UF1STC)	0000H
	LIN-UART1 receive data register (UF1RX)	0000H
	LIN-UART1 8-bit receive data register (UF1RXB)	00H
	LIN-UART1 transmit data register (UF1TX)	0000H
	LIN-UART1 8-bit transmit data register (UF1TXB)	00H
	LIN-UART1 wait transmit data register (UF1WTX)	0000H
	LIN-UART1 8-bit wait transmit data register (UF1WTXB)	00H
	LIN-UART1 ID setting register (UF1ID)	00H
	LIN-UART1 buffer registers 0, 1, 2, 3, 4, 5, 6, 7, 8 (UF1BUF1, UF1BUF1, UF1BUF2, UF1BUF3, UF1BUF4, UF1BUF5, UF1BUF6, UF1BUF7, UF1BUF8)	00H
	LIN-UART1 buffer control register (UF1BUCTL)	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.

Remark The special function register (SFR) mounted depend on the product. See **3.2.4 Special function registers (SFRs)** and **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 19-2. Hardware Statuses After Reset Acknowledgment (5/6)

Hardware		Status After Reset Acknowledgment[Note]
CAN controller	CAN global module control register (CGMCTRL)	0000H
	CAN global module clock select register (CGMCS)	0FH
	CAN global automatic block transmission control register (CGMABT)	0000H
	CAN global automatic block transmission delay setting register (CGMABTD)	00H
	CAN module mask 1 register (CMASK1L, CMASK1H)	Undefined
	CAN module mask 2 register (CMASK2L, CMASK2H)	Undefined
	CAN module mask 3 register (CMASK3L, CMASK3H)	Undefined
	CAN module mask 4 register (CMASK4L, CMASK4H)	Undefined
	CAN module control register (CCTRL)	0000H
	CAN module last error code register (CLEC)	00H
	CAN module information register (CINFO)	00H
	CAN module error counter register (CERC)	0000H
	CAN module interrupt enable register (CIE)	0000H
	CAN module interrupt status register (CINTS)	0000H
	CAN module bit rate prescaler register (CBRP)	FFH
	CAN module bit rate register (CBTR)	370FH
	CAN module last in-pointer register (CLIPT)	Undefined
	CAN module receive history list register (CRGPT)	xx02H
	CAN module last out-pointer register (CLOPT)	Undefined
	CAN module transmit history list register (CTGPT)	xx02H
	CAN module time stamp register (CTS)	0000H
Multiplier/divider	Multiplication/division data register A (L) (MDAL)	0000H
	Multiplication/division data register A (H) (MDAH)	0000H
	Multiplication/division data register B (L) (MDBL)	0000H
	Multiplication/division data register B (H) (MDBH)	0000H
	Multiplication/division data register C (L) (MDCL)	0000H
	Multiplication/division data register C (H) (MDCH)	0000H
	Multiplication/division control register (MDUC)	00H
Key interrupt	Key return mode register (KRM)	00H
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
	DMA all-channel forced wait register (DMCALL)	00H

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Remark The special function register (SFR) mounted depend on the product. See **3.2.4 Special function registers (SFRs)** and **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 19-2. Hardware Statuses After Reset Acknowledgment (6/6)

Hardware		Status After Reset Acknowledgment[Note 1]
Reset function	Reset control flag register (RESF)	00H[Note 2]
	POC reset register (POCRES)	00H[Note 3]
Low-voltage detector	Low-voltage detection register (LVIM)	00H/82H[Note 4]
<R>	Low-voltage detection level select register (LVIS)	09H[Note 2]
Interrupt	Request flag registers 0L, 0H, 1L, 1H, 2L, 2H, 3L, 3H (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H)	00H
	Mask flag registers 0L, 0H, 1L, 1H, 2L, 2H, 3L, 3H (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H)	FFH
	Priority specification flag registers 00L, 00H, 01L, 01H, 02L, 02H, 03L, 03H, 10L, 10H, 11L, 11H, 12L, 12H, 13L, 13H (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H)	FFH
	External interrupt rising edge enable registers 0, 1 (EGP0, EGP1)	00H
	External interrupt falling edge enable registers 0, 1 (EGN0, EGN1)	00H
BCD correction circuit	BCD correction result register (BCDAJ)	Undefined

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 clear source.

Register \ Clear Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by RESF Register Read	Reset by WDT	Reset by CLM	Reset by Illegal Memory Access	Reset by LVI
RESF	TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Cleared (0)	Held	Held	Held	Held
	WDRF bit			Held		Set (1)	Held	Held	Held
	CLKRF bit			Held		Held	Set (1)	Held	Held
	IAWRF bit			Held		Held	Held	Set (1)	Held
	LVIRF bit			Held		Held	Held	Held	Set (1)
POCRES	POCRES_0	Held		Held	Held	Held	Held	Held	Held
LVIS		Cleared (09H)	Cleared (09H)	Cleared (09H)	Held	Cleared (09H)	Cleared (09H)	Cleared (09H)	Held
LVIM		Cleared (00H/82H[Note 4])	Cleared (00H/82H[Note 4])	Cleared (00H/82H[Note 4])	Held	Cleared (00H/82H[Note 4])	Cleared (00H/82H[Note 4])	Cleared (00H/82H[Note 4])	Held

3. When a reset by a source other than POC is generated, the value immediately before the reset is retained.

4. This value varies depending on the reset source and the option byte.

Remark The special function register (SFR) mounted depend on the product. See **3.2.4 Special function registers (SFRs)** and **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

19.1 Register for Confirming Reset Source

(1) Reset Control Flag Register (RESF)

Many internal reset generation sources exist in the 78K0R/Hx3. 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 set RESF to 00H.

Cautions 1. RESF also becomes 00H when $\overline{\text{RESET}}$ is input simultaneously during power application.

2. RESF also becomes 00H if the $\overline{\text{RESET}}$ pin was used before starting the CPU operation after the POC reset was released.

Figure 19-5. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: 00H[Note 1] R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDRF	0	CLKRF	IAWRF	LVIRF

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.

CLKRF	Internal reset request by clock monitor
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

IAWRF	Internal reset request by illegal memory access
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-chipdebug emulator.

Cautions 1. Do not read data by a 1-bit memory manipulation instruction.

2. 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 19-3.

Table 19-3. RESF Status When Reset Request Is Generated

Clear Source Flag	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by RESF Read	Reset by WDT	Reset by CLM	Reset by Illegal Memory Access	Reset by LVI
TRAP	Cleared (0)	Cleared (0)	Set (1)	Cleared (0)	Held	Held	Held	Held
WDRF			Held		Set (1)	Held	Held	Held
CLKRF			Held		Held	Set (1)	Held	Held
IAWRF			Held		Held	Held	Set (1)	Held
LVIRF			Held		Held	Held	Held	Set (1)

<R> **(2) POC reset register (POCRES)**

The POC reset register (POCRES) checks the generation of a POC reset.
With POCRES, only writing “1” is valid and writing “0” is invalid.
Only a reset by the power-on-clear (POC) circuit clears this register to 00H.

Caution When POCRES = 1, it is guaranteed that no POC reset will be generated, but the RAM value is not guaranteed.

Remark Preset POCRES_0 to “1” when checking a reset by the power-on-clear (POC) circuit.

Figure 19-6. Format of POC Reset Register (POCRES)

Address: F00FBH After reset: 00H[Note] R/W

Symbol	7	6	5	4	3	2	1	0
POCRES	0	0	0	0	0	0	0	POCRES_0

POCRES_0	Internal reset request by POC reset
0	A POC reset is generated or writing is not performed.
1	A POC reset is not generated.

Note When a reset by a source other than POC is generated, the value immediately before the reset is retained.

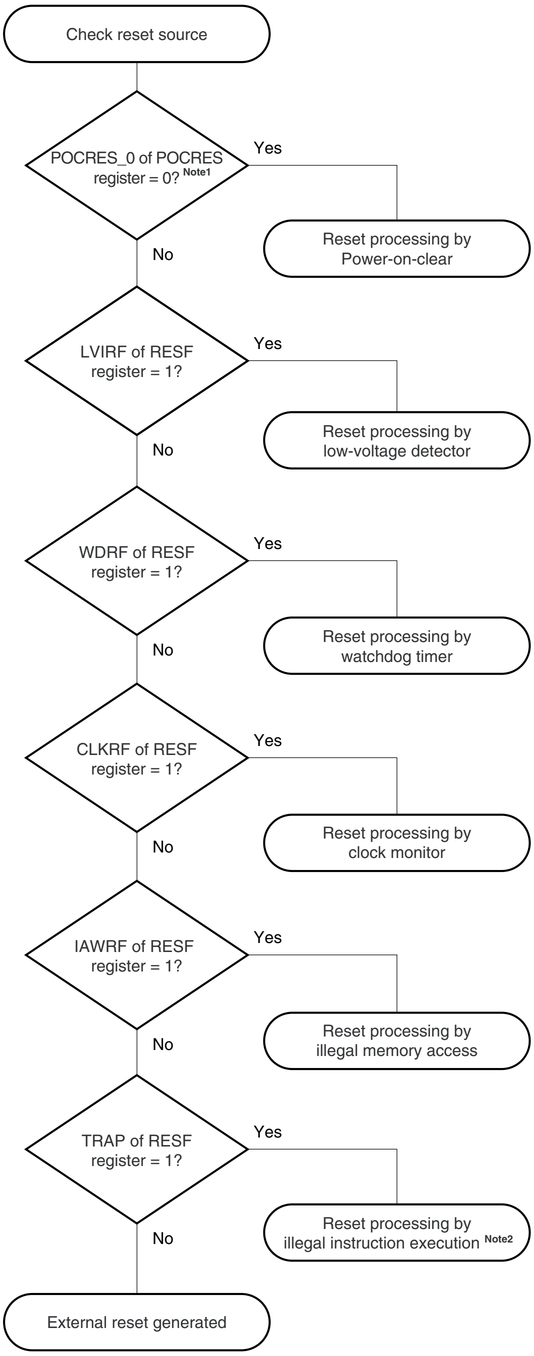
Table 19-4 shows the POCRES state when a reset is requested.

Table 19-4. POCRES Status When Reset Request Is Generated

Clear Source Flag	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by RESF Read	Reset by WDT	Reset by CLM	Reset by Illegal Memory Access	Reset by LVI
POCRES_0	Held	Cleared (0)	Held	Held	Held	Held	Held	Held

Example of software processing after reset release is shown below.

Figure 19-7. Example of Software Processing After Reset Release


Check reset source
POCRES_0 of POCRES register = 0? Note1
Yes
No
Reset processing by Power-on-clear
LVIRF of RESF register = 1?
Yes
No
Reset processing by low-voltage detector
WDRF of RESF register = 1?
Yes
No
Reset processing by watchdog timer
CLKRF of RESF register = 1?
Yes
No
Reset processing by clock monitor
IAWRF of RESF register = 1?
Yes
No
Reset processing by illegal memory access
TRAP of RESF register = 1?
Yes
No
Reset processing by illegal instruction execution Note2
External reset generated

Notes 1. Preset POCRES_0 to “1” when detecting a reset by the power-on-clear (POC) circuit.

2. 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 multiple reset sources occur, the corresponding source flags will be set at the same time.

CHAPTER 20 POWER-ON-CLEAR CIRCUIT

20.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) 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 2.93 V±0.2 V.

<R>
- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 V), generates internal reset signal when $V_{DD} < V_{PDR}$.

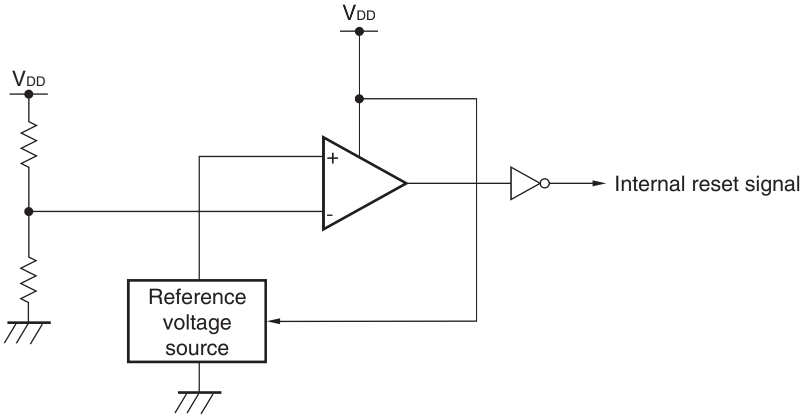
  Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

  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), illegal instruction execution, clock monitor, or illegal memory access. RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT, LVI, illegal instruction execution, clock monitor, or illegal memory access.
  For details of RESF, see **CHAPTER 19 RESET FUNCTION**.

20.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 20-1.

<R> **Figure 20-1. Block Diagram of Power-on-Clear Circuit**



<R> 20.3 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_{POR} = 1.61 V ±0.09 V), the reset status is released.

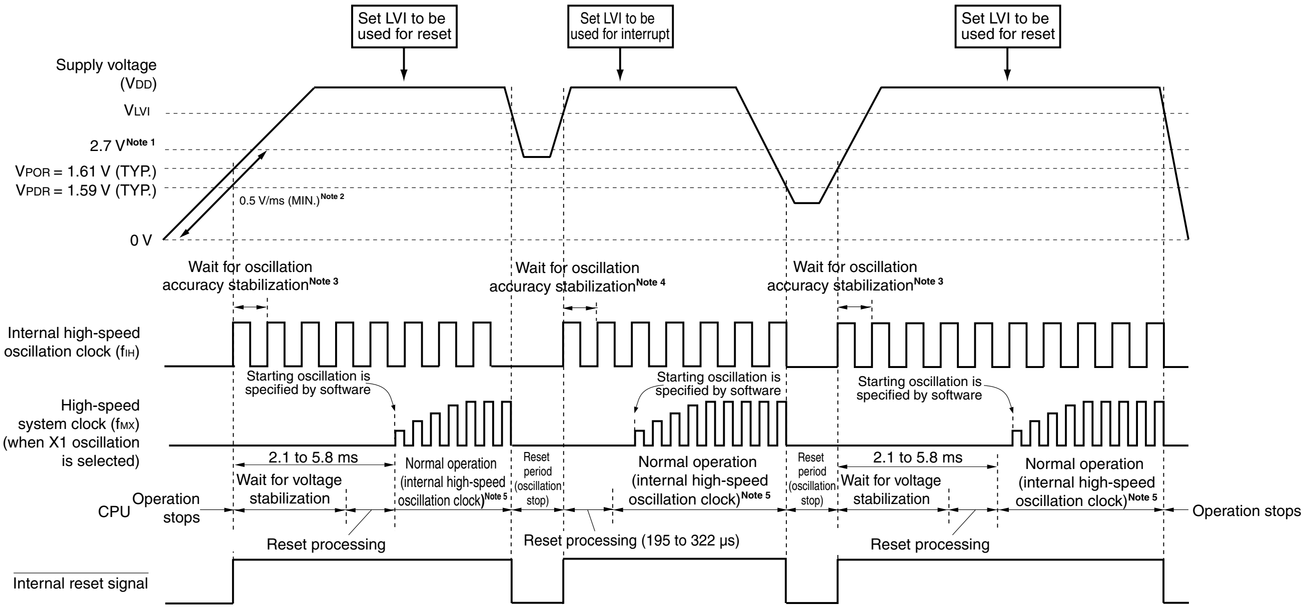
  Caution If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage (V_{DD}) exceeds 2.93 V±0.2 V.

- 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 20-2. 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)


Set LVI to be used for reset
Set LVI to be used for interrupt
Set LVI to be used for reset
Supply voltage (VDD)
VLVI
2.7 V Note 1
VPOR = 1.61 V (TYP.)
VPDR = 1.59 V (TYP.)
0 V
0.5 V/ms (MIN.) Note 2
Wait for oscillation accuracy stabilization Note 3
Wait for oscillation accuracy stabilization Note 4
Wait for oscillation accuracy stabilization Note 3
Internal high-speed oscillation clock (fIH)
Starting oscillation is specified by software
Starting oscillation is specified by software
Starting oscillation is specified by software
High-speed system clock (fMX) (when X1 oscillation is selected)
2.1 to 5.8 ms
Wait for voltage stabilization
Normal operation (internal high-speed oscillation clock) Note 5
Reset period (oscillation stop)
Normal operation (internal high-speed oscillation clock) Note 5
Reset period (oscillation stop)
2.1 to 5.8 ms
Wait for voltage stabilization
Normal operation (internal high-speed oscillation clock) Note 5
CPU
Operation stops
Operation stops
Reset processing
Reset processing (195 to 322 μs)
Reset processing
Internal reset signal

Notes 1. The operation guaranteed range is 2.7 V ≤ VDD ≤ 5.5 V. Make sure to perform normal operation after the supply voltage has become at least 2.7 V. To make the state at lower than 2.7 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the RESET pin.

2. If the rate at which the voltage rises to 2.7 V after power application is slower than 0.5 V/ms (MIN.), input a low level to the RESET pin or use the option byte to set the LVI to be on (option byte: LVIOFF = 0) by default.

3. The internal voltage stabilization time 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 21 LOW-VOLTAGE DETECTOR).

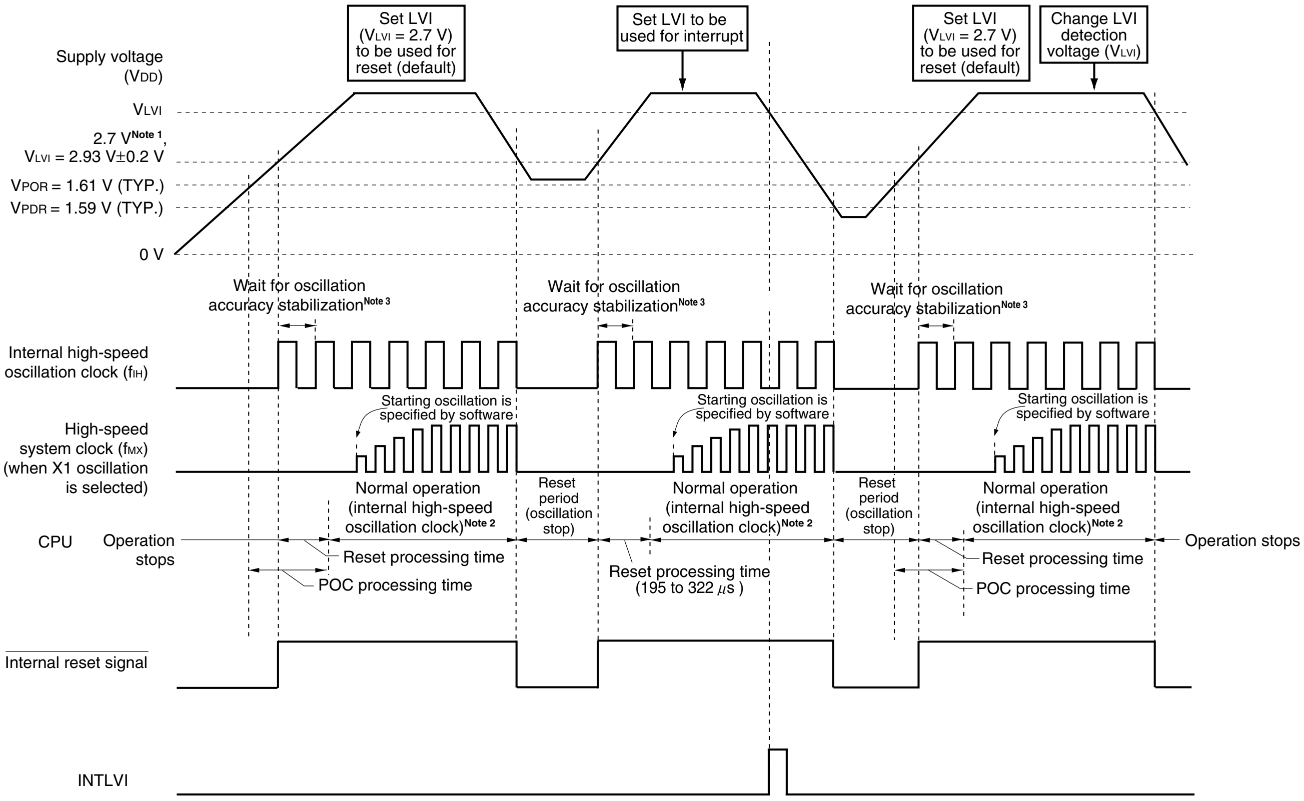
Remark V_{LVI}: LVI detection voltage

V_{POR}: POC power supply rise detection voltage

V_{PDR}: POC power supply fall detection voltage

Figure 20-2. 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)


Set LVI (VLVI = 2.7 V) to be used for reset (default)
Set LVI to be used for interrupt
Set LVI (VLVI = 2.7 V) to be used for reset (default)
Change LVI detection voltage (VLVI)
Supply voltage (VDD)
VLVI
2.7 V Note 1, VLVI = 2.93 V±0.2 V
VPOR = 1.61 V (TYP.)
VPDR = 1.59 V (TYP.)
0 V
Wait for oscillation accuracy stabilization Note 3
Internal high-speed oscillation clock (fIH)
Starting oscillation is specified by software
High-speed system clock (fMX) (when X1 oscillation is selected)
Normal operation (internal high-speed oscillation clock) Note 2
Reset period (oscillation stop)
CPU
Operation stops
Reset processing time
POC processing time
Reset processing time (195 to 322 μs)
Internal reset signal
INTLVI

Notes 1. The operation guaranteed range is 2.7 V ≤ V_{DD} ≤ 5.5 V. Make sure to perform normal operation after the supply voltage has become at least 2.7 V. To make the state at lower than 2.7 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 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.

3. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 21 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

20.4 Cautions for Power-on-Clear Circuit

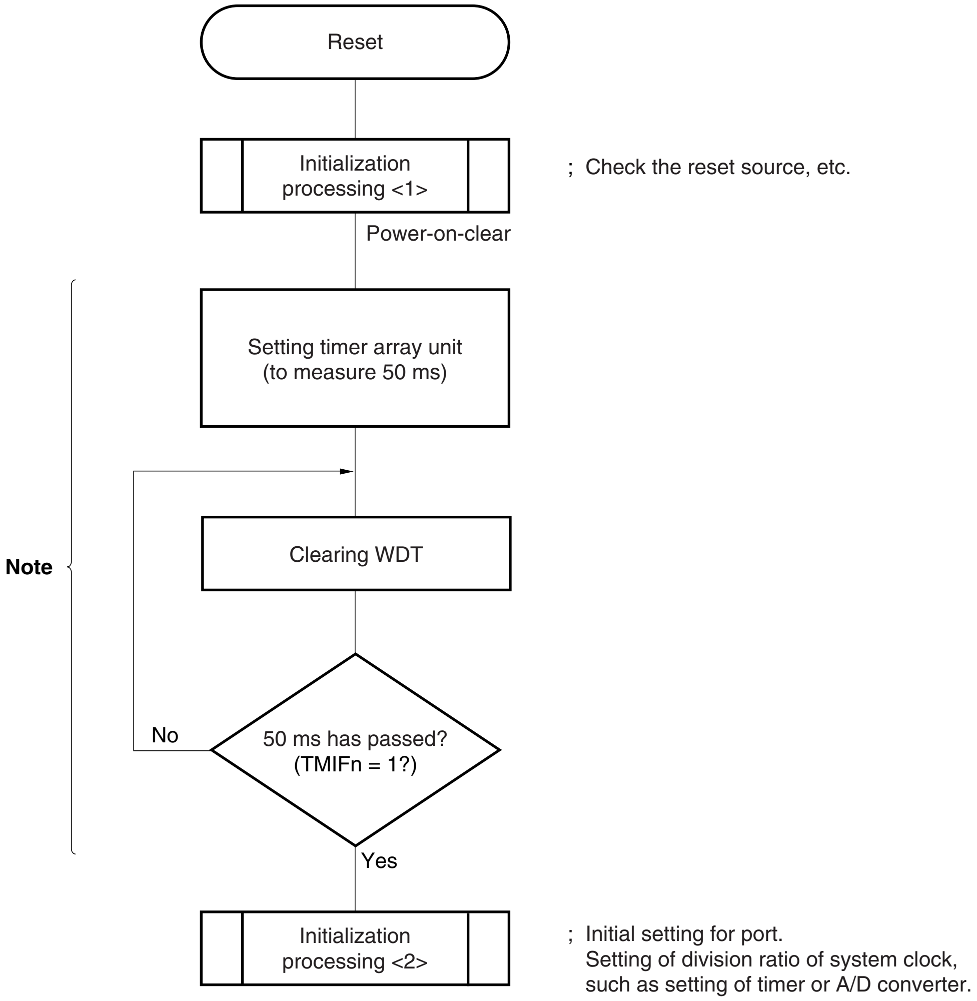
In a system where the supply voltage (V_{POR}, V_{PDR}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller 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 20-3. Example of Software Processing After Reset Release

- If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage


Reset
Initialization processing <1>
; Check the reset source, etc.
Power-on-clear
Setting timer array unit (to measure 50 ms)
Note
Clearing WDT
No
50 ms has passed? (TMIFn = 1?)
Yes
Initialization processing <2>
; Initial setting for port.
Setting of division ratio of system clock,
such as setting of timer or A/D converter.

Note If reset is generated again during this period, initialization processing <2> is not started.

Remark n = 00 to 07, 10 to 17: 78K0R/HC3,
n = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3, n = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

CHAPTER 21 LOW-VOLTAGE DETECTOR

21.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) 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[Note] 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[Note] is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.93 V ±0.2 V). After that, the internal reset signal[Note] is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.84 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}, 10 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

Note See the timing in **Figure 20-2 (2) When LVI is ON upon power application (option byte: LVIOFF = 0)** for the reset processing time until the normal operation is entered after the LVI reset is released.

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

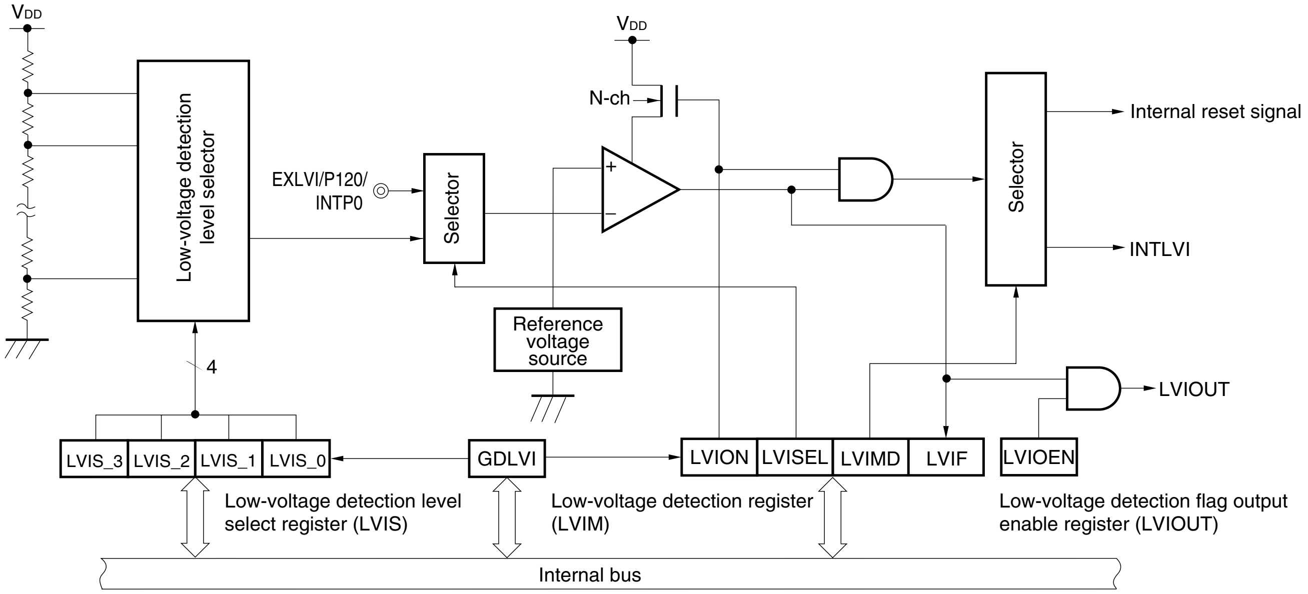
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

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 19 RESET FUNCTION**.

21.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 21-1.

Figure 21-1. Block Diagram of Low-Voltage Detector


VDD
Low-voltage detection level selector
4
EXLVI/P120/ INTP0
Selector
VDD
N-ch
+
−
Reference voltage source
Selector
Internal reset signal
INTLVI
LVIOUT
LVIS_3
LVIS_2
LVIS_1
LVIS_0
GDLVI
LVION
LVISEL
LVIMD
LVIF
LVIOEN
Low-voltage detection level select register (LVIS)
Low-voltage detection register (LVIM)
Low-voltage detection flag output enable register (LVIOUT)
Internal bus

21.3 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)
- Low-voltage detection flag output enable register (LVIOUT)

(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.

Caution Writing to the LVIM register is valid only when bit 5 (GDLVI bit) of the special register manipulation protect register (GUARD) is 1.

Figure 21-2. Format of Low-Voltage Detection Register (LVIM)

Address: FFFA9H After reset: 00H/82H[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[Note4]	Voltage detection selection
0	Detects level of supply voltage (V_{DD})
1	Detects level of input voltage from external input pin (EXLVI)

LVIMD[Note4]	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[Note4]	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. When the LVION bit 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 the LVION bit is set to 1 and when the voltage is confirmed with LVIF flag.

- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))

The LVIF flag 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.

4. It takes one clock cycle to apply changes to the settings of the LVIM register. After changing the settings, wait at least one clock cycle before reading the LVIM register.

(Cautions 1 to 3 are given on the next page.)

Cautions 1. To stop LVI, follow the procedure below.

- **When using 1-bit memory manipulation instruction: Clear LVION to 0.**

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.

<R> **(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 09H.

Caution Writing to the LVIS register is valid only when bit 5 (GDLVI bit) of the special register manipulation protect register (GUARD) is 1.

Figure 21-3. Format of Low-Voltage Detection Level Select Register (LVIS)

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

Symbol	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS_3	LVIS_2	LVIS_1	LVIS_0

LVIS_3	LVIS_2	LVIS_1	LVIS_0	Detection level
0	0	0	0	V_{LVI0} (4.22 ±0.1 V)
0	0	0	1	V_{LVI1} (4.07 ±0.1 V)
0	0	1	0	V_{LVI2} (3.92 ±0.1 V)
0	0	1	1	V_{LVI3} (3.76 ±0.1 V)
0	1	0	0	V_{LVI4} (3.61 ±0.1 V)
0	1	0	1	V_{LVI5} (3.45 ±0.1 V)
0	1	1	0	V_{LVI6} (3.30 ±0.1 V)
0	1	1	1	V_{LVI7} (3.15 ±0.1 V)
1	0	0	0	V_{LVI8} (2.99 ±0.1 V)
1	0	0	1	V_{LVI9} (2.84 ±0.1 V)
Other than 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 “09H” if a reset other than by LVI is effected.

Cautions 1. Be sure to clear bits 4 to 7 to “0”.

2. Do not change the value of LVIS while the LVI operates (LVION = 1).

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.

<R> **(3) Port mode register 12 (PM12)**

When using the P120/EXLVI/INTP0 pin for external low-voltage detection potential input, set PM12_0 to 1. At this time, the output latch of P12_0 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 21-4. Format of Port Mode Register 12 (PM12)

Address: FFF2CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM12	PM12_7	PM12_6	PM12_5	1	1	1	1	PM12_0

PM12_0	P120 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark When EXLVI is used, 0 is read from P12_0.

(4) Low-voltage detection flag output enable register (LVIOUT)

his register is used to select whether to output the value of the low-voltage detection flag (LVIM.LVIF) from the P70/LVIOUT/INTP5/KR0/TI15/TO15 pin.

This register is initialized by all types of resets. When a reset occurs, P70 becomes high impedance, so this register can only be used when “interrupt” is selected as the operating mode (LVIM.LVIMD = 0).

Caution This register is not protected by the GUARD register.

Figure 21-5. Format of Low-voltage Detection Flag Output Enable Register (LVIOUT)

Address: F0067H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LVIOUT	0	0	0	0	0	0	0	LVIOEN

LVIOEN	Control of low-voltage detection flag (LVIM.LVIF) output
0	The value of the low-voltage detection flag (LVIM.LVIF) is not output from a pin.
1	The value of the low-voltage detection flag (LVIM.LVIF) is output from P70.

Note When using this register, set P70 to output mode (PM7_0 = 0) and set the port latch (P7_0) to 0. Also, do not use any alternate functions other than LVIOUT of P70 pin.

21.4 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} = 1.21 V ±0.1 V), 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} = 2.93 V ±0.2 V). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.84 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

21.4.1 When used as reset

(1) When detecting level of supply voltage (V_{DD})

(a) When LVI Default Start Function Stopped Is Set (Option Byte: 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 (LVIS_3 to LVIS_0) 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 21-6 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.

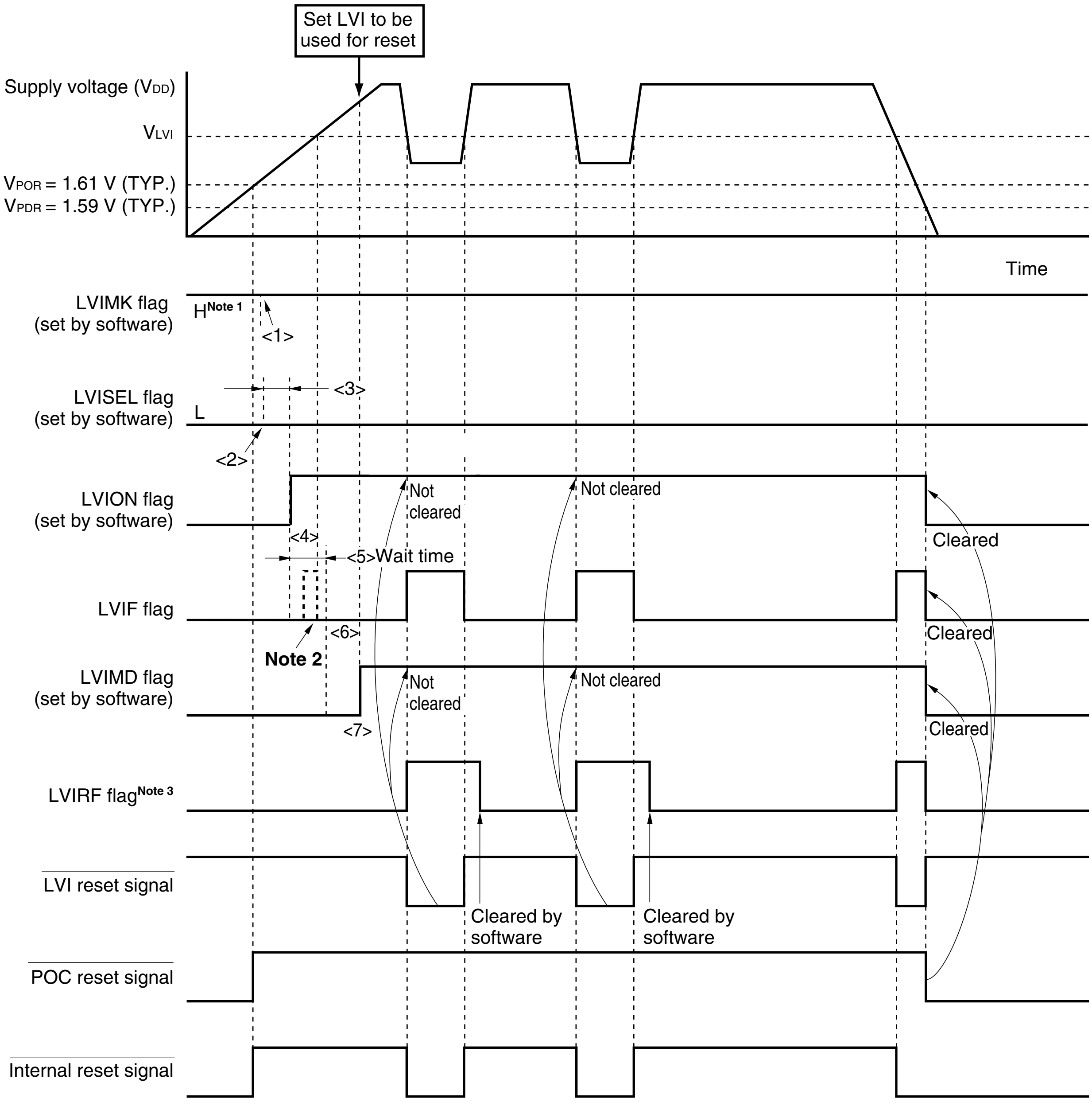
Cautions 1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <4>.

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**

Clear (0) the LVIMD bit and then the LVION bit by using a 1-bit memory manipulation instruction.

Figure 21-6. 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 19 RESET FUNCTION**.

Remarks 1. <1> to <7> in Figure 21-6 above correspond to <1> to <7> in the description of “When starting operation” in **21.4.1 (1) (a) When LVI Default Start Function Stopped Is Set (Option Byte: 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 (Option Byte: 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 09H (default value: V_{LVI} = 2.93 V ±0.2 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}) $\geq$ detection voltage (V_{LVI})”)

  Figure 21-6 shows the timing of the internal reset signal generated by the low-voltage detector.

- When stopping operation

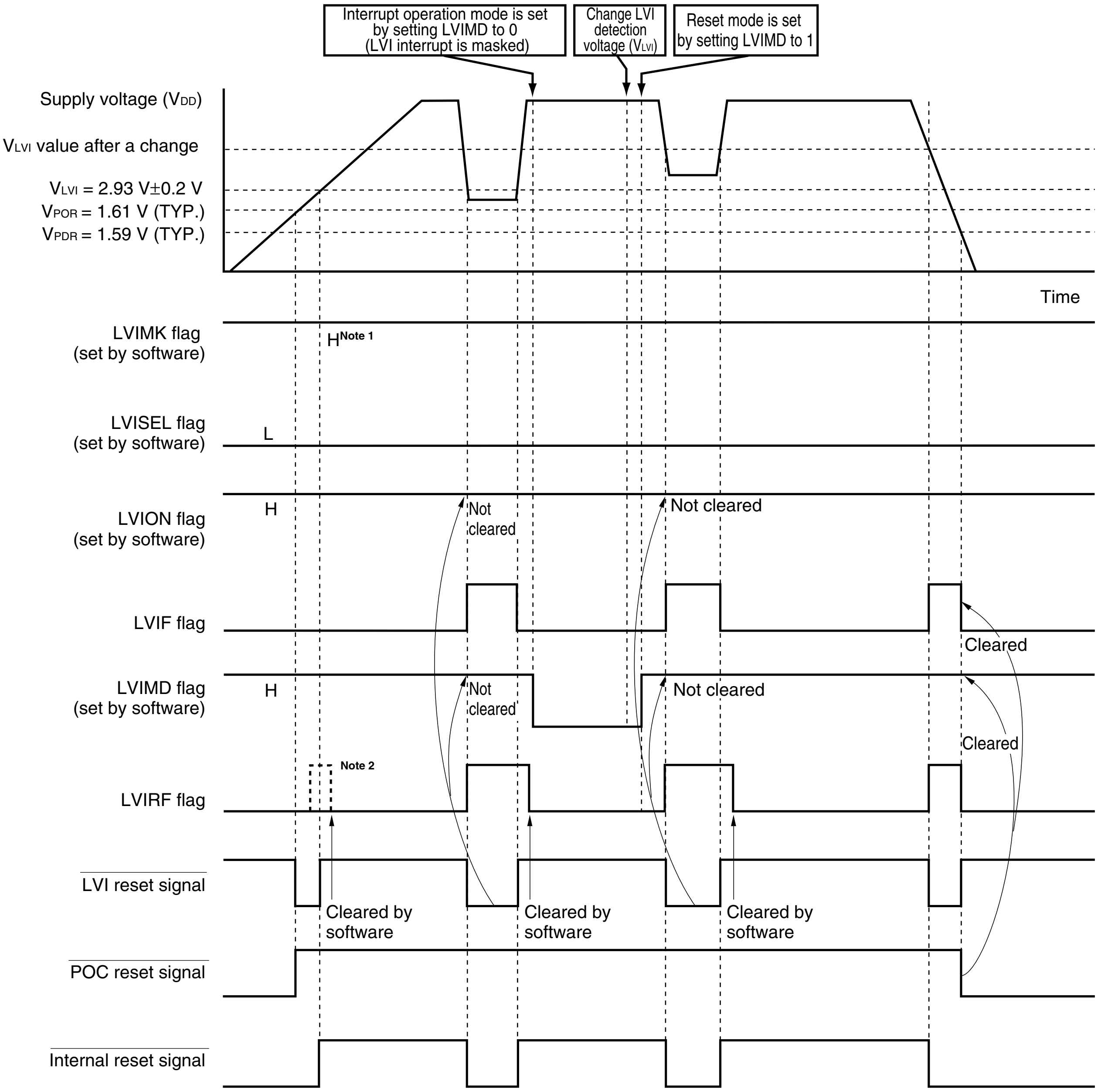
  Clear (0) the LVIMD bit and then the LVION bit by using a 1-bit memory manipulation instruction.

  Figure 21-8 shows the internal reset signal generation timing when LVI operation is stopped.

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. If a reset other than POC or a pin reset occurs, however, there is a period when low-voltage detection cannot be performed normally.**

Figure 21-7. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 0)


Interrupt operation mode is set by setting LVIMD to 0 (LVI interrupt is masked)
Change LVI detection voltage (VLVI)
Reset mode is set by setting LVIMD to 1
Supply voltage (VDD)
VLVI value after a change
VLVI = 2.93 V±0.2 V
VPOR = 1.61 V (TYP.)
VPDR = 1.59 V (TYP.)
Time
LVIMK flag (set by software)
H Note 1
LVISEL flag (set by software)
L
LVION flag (set by software)
H
Not cleared
Not cleared
LVIF flag
Cleared
LVIMD flag (set by software)
H
Not cleared
Not cleared
Cleared
Note 2
LVIRF flag
LVI reset signal
Cleared by software
Cleared by software
Cleared by software
POC reset signal
Internal reset signal

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).

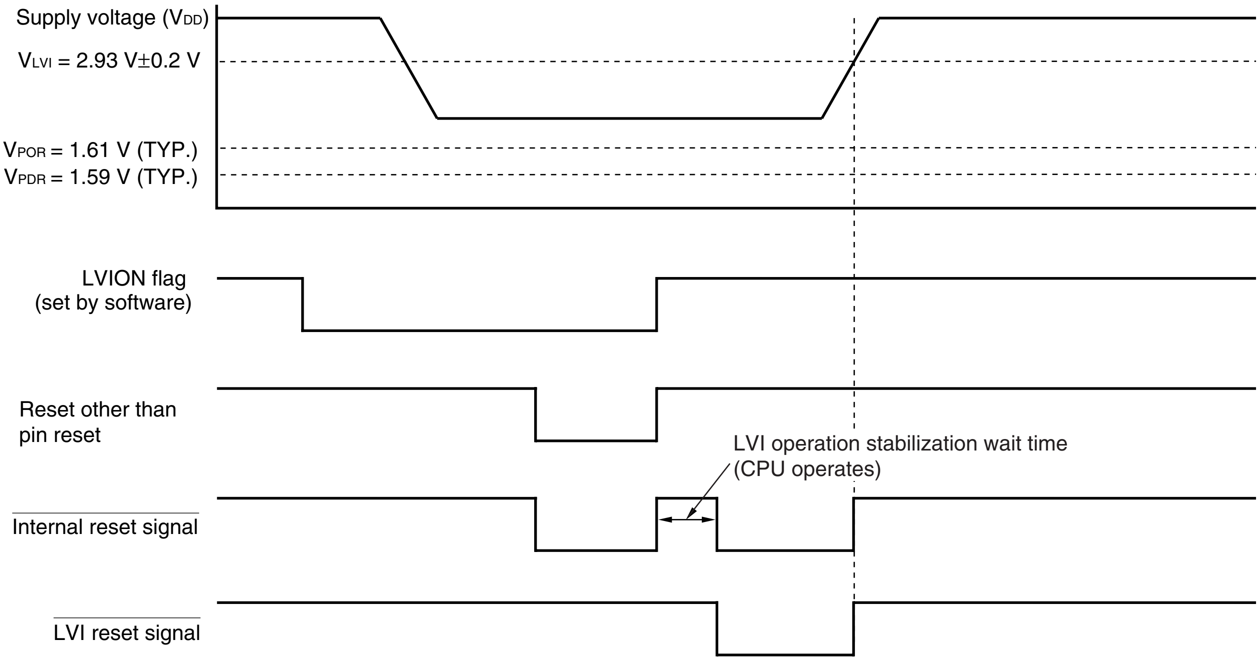
When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, the LVIRF flag may become 1 from the beginning due to the power-on waveform.

For details of RESF, see **CHAPTER 19 RESET FUNCTION**.

Remark V_{POR}: POC power supply rise detection voltage

V_{PDR}: POC power supply fall detection voltage

Figure 21-8. Internal Reset Signal Generation Timing When LVI Operation Stop Is Set (Bit: LVISEL = 0, Option Byte: LVIOFF = 0)



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 21-9 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. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.

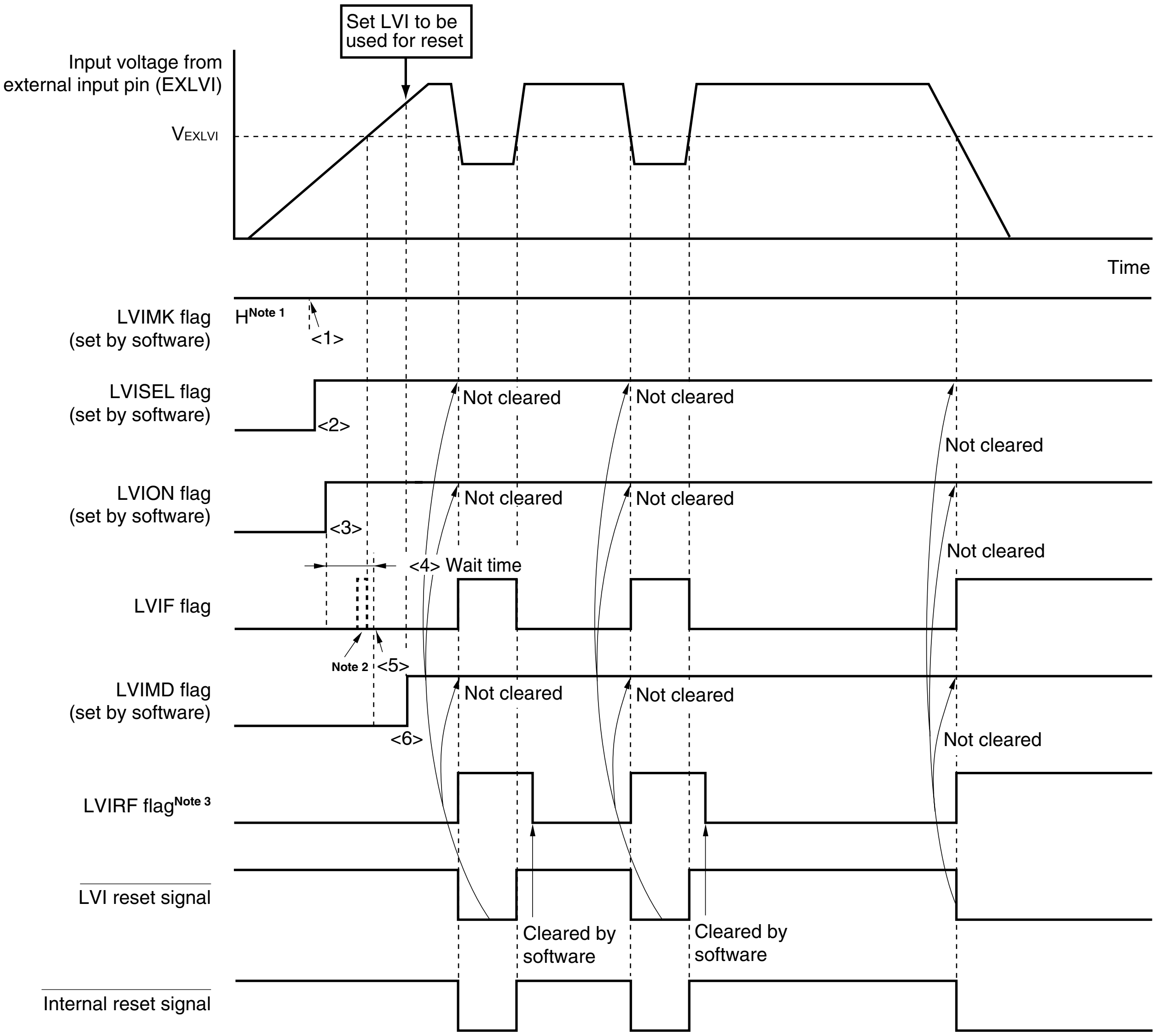
  2. If input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.

  <R> **3. Input voltage from external input pin (EXLVI) must be EXLVI ≤ V_{DD}.**

- When stopping operation

  Clear (0) the LVIMD bit and then the LVION bit by using a 1-bit memory manipulation instruction.

Figure 21-9. 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 19 RESET FUNCTION**.

Remark <1> to <6> in Figure 21-9 above correspond to <1> to <6> in the description of “When starting operation” in **21.4.1 (2) When detecting level of input voltage from external input pin (EXLVI)**.

21.4.2 When used as interrupt

(1) When detecting level of supply voltage (V_{DD})

(a) When LVI Default Start Function Stopped Is Set (Option Byte: 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 (LVIS_3 to LVIS_0) 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).

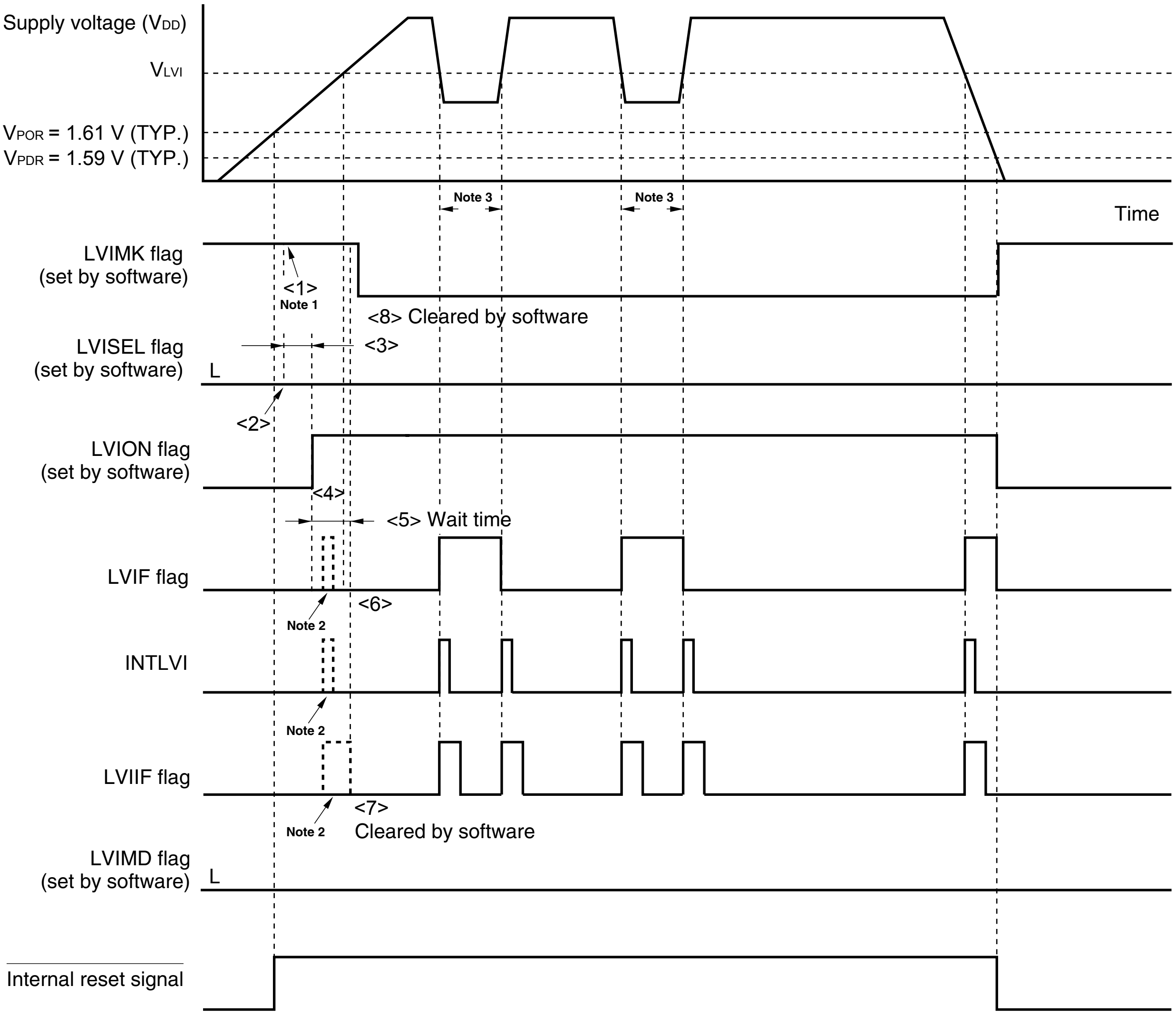
<9> Execute the EI instruction (when vector interrupts are used).

Figure 21-10 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

Clear (0) the LVION bit by using a 1-bit memory manipulation instruction.

Figure 21-10. 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 21-10 above correspond to <1> to <8> in the description of “When starting operation” in **21.4.2 (1) (a) When LVI Default Start Function Stopped Is Set (Option Byte: 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 (Option Byte: 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 09H (default value: V_{LVI} = 2.93 V ±0.2 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 21-11 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

  Clear (0) the LVION bit by using a 1-bit memory manipulation instruction.

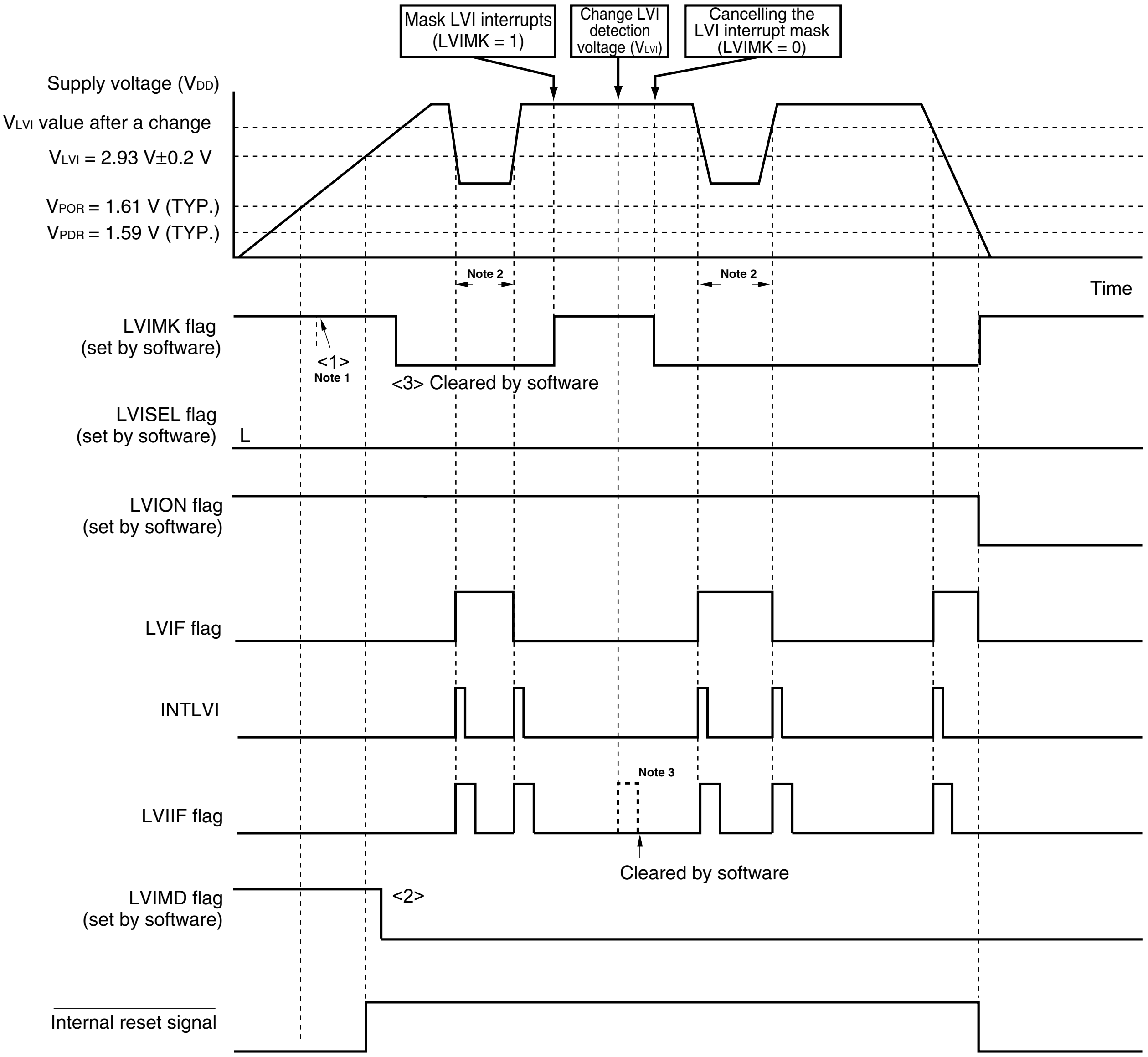
  Figure 21-12 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <3> above.

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. If a reset other than POC or a pin reset occurs, however, there is a period when low-voltage detection cannot be performed normally.**

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 19 RESET FUNCTION.

Figure 21-11. Timing 1 of Low-Voltage Detector Interrupt Signal Generation
(Bit: LVISEL = 0, Option Byte: LVIOFF = 0)


Mask LVI interrupts (LVIMK = 1)
Change LVI detection voltage (V_{LVI})
Cancelling the LVI interrupt mask (LVIMK = 0)
Supply voltage (V_{DD})
V_{LVI} value after a change
V_{LVI} = 2.93 V±0.2 V
V_{POR} = 1.61 V (TYP.)
V_{PDR} = 1.59 V (TYP.)
Note 2
Note 2
Time
LVIMK flag (set by software)
<1>
Note 1
<3> Cleared by software
LVISEL flag (set by software)
L
LVION flag (set by software)
LVIF flag
INTLVI
Note 3
LVIIF flag
Cleared by software
LVIMD flag (set by software)
<2>
Internal reset signal

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.

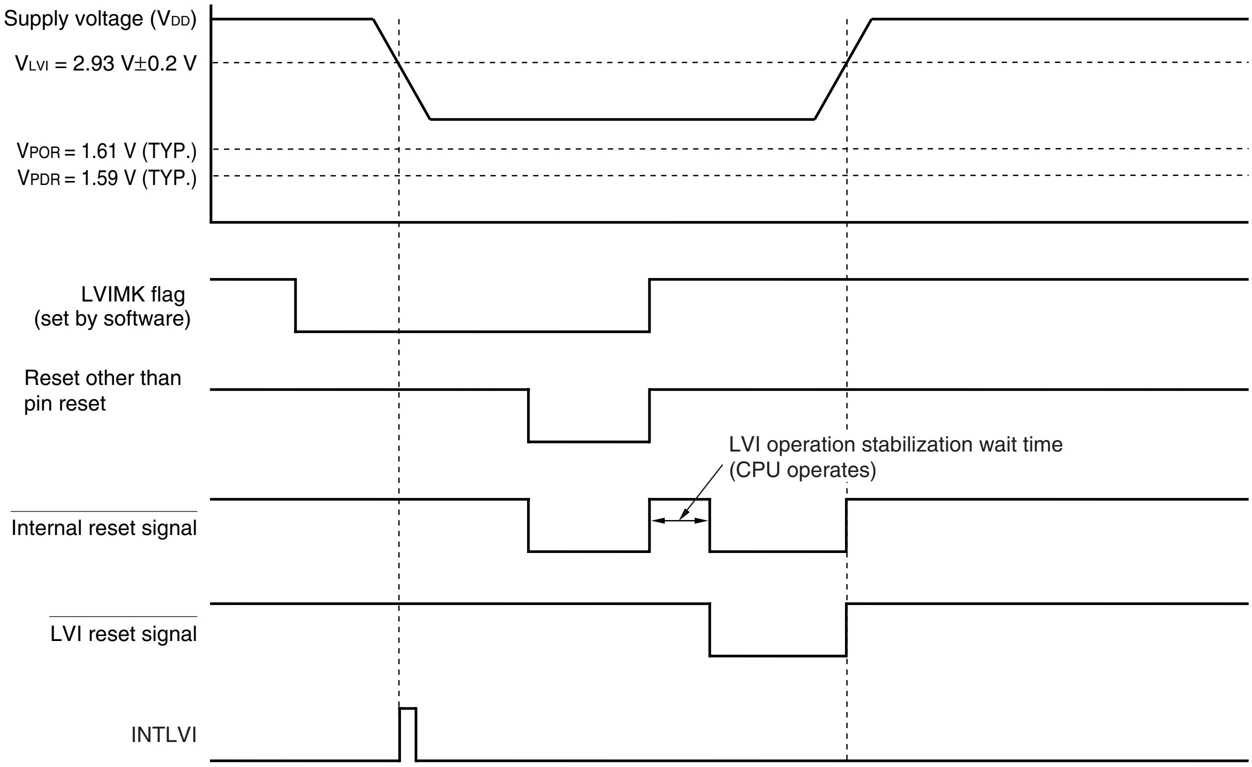
3. The LVIIF flag may be set when the LVI detection voltage is changed.

Remarks 1. <1> to <3> in Figure 21-11 above correspond to <1> to <3> in the description of “When starting operation” in **21.4.2 (1) (b) When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0).**

2. V_{POR}: POC power supply rise detection voltage

V_{PDR}: POC power supply fall detection voltage

Figure 21-12. Timing 2 of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 0)



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)).
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).

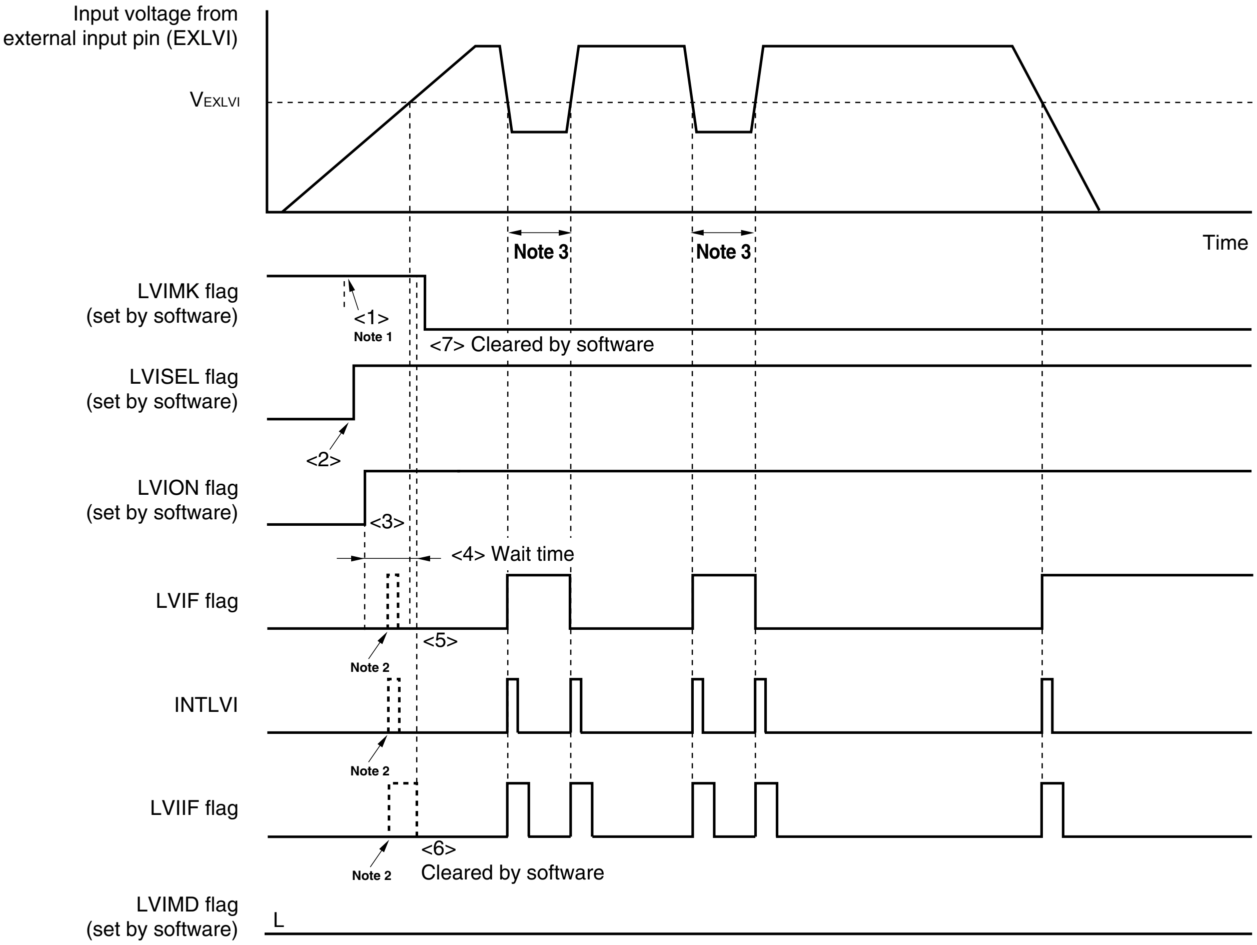
Figure 21-13 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

Caution Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation

Clear (0) the LVION bit by using a 1-bit memory manipulation instruction.

Figure 21-13. Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 1)


Input voltage from external input pin (EXLVI)
VEXLVI
Time
Note 3
Note 3
LVIMK flag (set by software)
<1>
Note 1
<7> Cleared by software
LVISEL flag (set by software)
<2>
LVION flag (set by software)
<3>
<4> Wait time
LVIF flag
<5>
Note 2
INTLVI
Note 2
LVIIF flag
<6>
Cleared by software
Note 2
LVIMD flag (set by software)
L

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 21-13 above correspond to <1> to <7> in the description of “When starting operation” in **21.4.2 (2) When detecting level of input voltage from external input pin (EXLVI)**.

21.5 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>

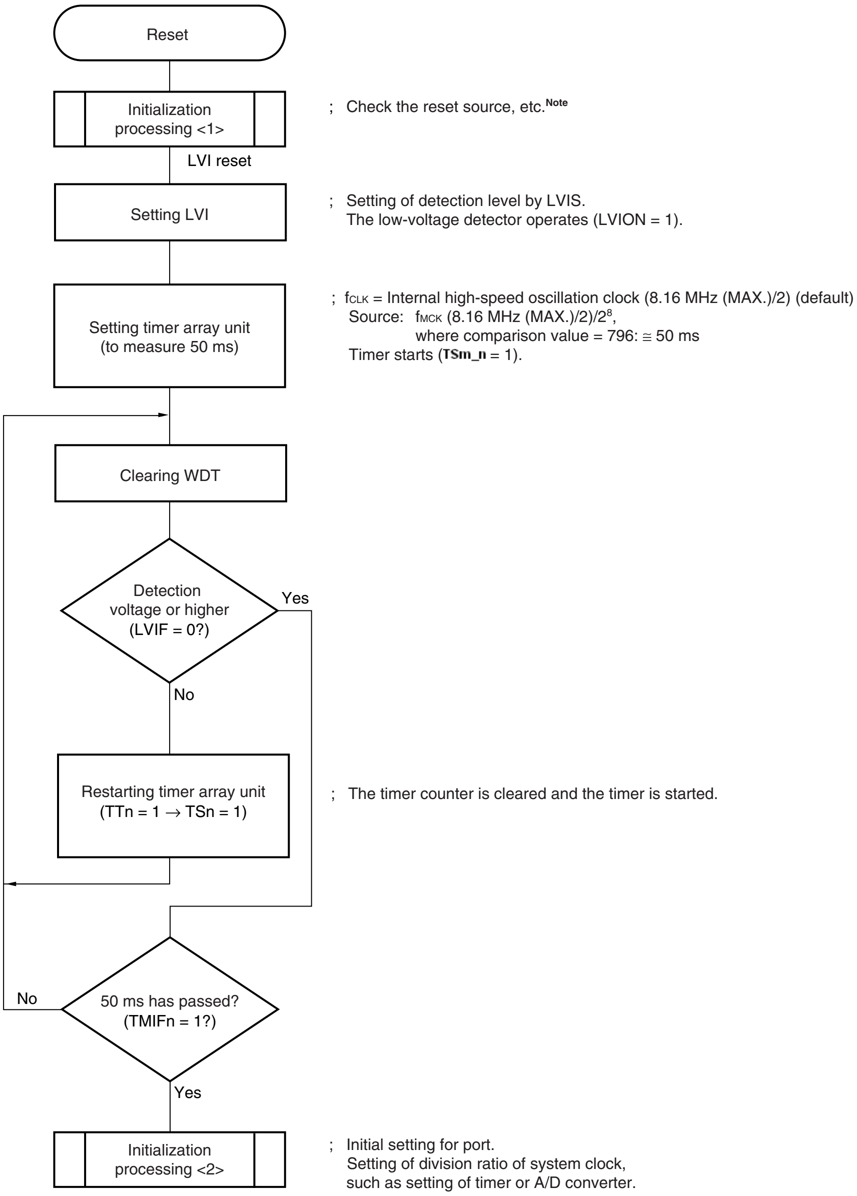
After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (see **Figure 21-14**).

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 21-14. Example of Software Processing After Reset Release (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage


Reset
Initialization processing <1>
LVI reset
Setting LVI
Setting timer array unit (to measure 50 ms)
Clearing WDT
Detection voltage or higher (LVIF = 0?)
Yes
No
Restarting timer array unit (TTn = 1 → TSn = 1)
50 ms has passed? (TMIFn = 1?)
No
Yes
Initialization processing <2>

; Check the reset source, etc.[Note]

; Setting of detection level by LVIS.
The low-voltage detector operates (LVION = 1).

; f_{CLK} = Internal high-speed oscillation clock (8.16 MHz (MAX.)/2) (default)
Source: f_{MCK} (8.16 MHz (MAX.)/2)/2^8,
where comparison value = 796: ≅ 50 ms
Timer starts (TSm_n = 1).

; The timer counter is cleared and the timer is started.

; Initial setting for port.
Setting of division ratio of system clock,
such as setting of timer or A/D converter.

Note A flowchart is shown on the next page.

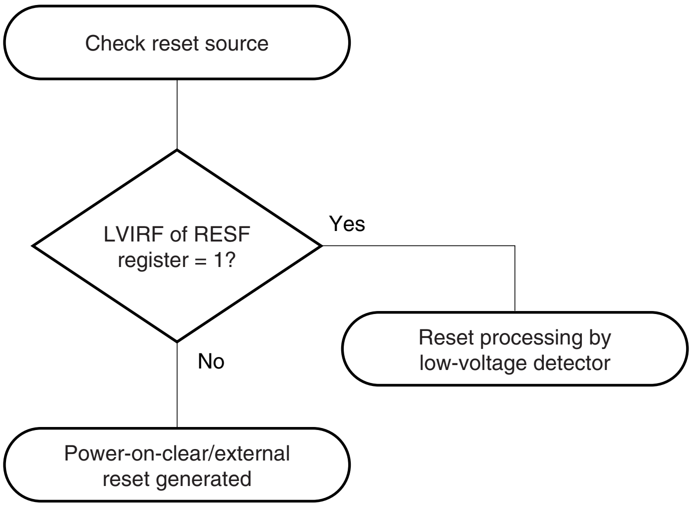
Remarks 1. If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to “1”, the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

2. n = 00 to 07, 10 to 17: 78K0R/HC3,
n = 00 to 07, 10 to 17, 20 to 23: 78K0R/HE3, 78K0R/HF3,
n = 00 to 07, 10 to 17, 20 to 27: 78K0R/HG3

Figure 21-14. Example of Software Processing After Reset Release (2/2)

- Checking reset source


Check reset source
LVIRF of RESF register = 1?
Yes
No
Reset processing by low-voltage detector
Power-on-clear/external reset generated

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)

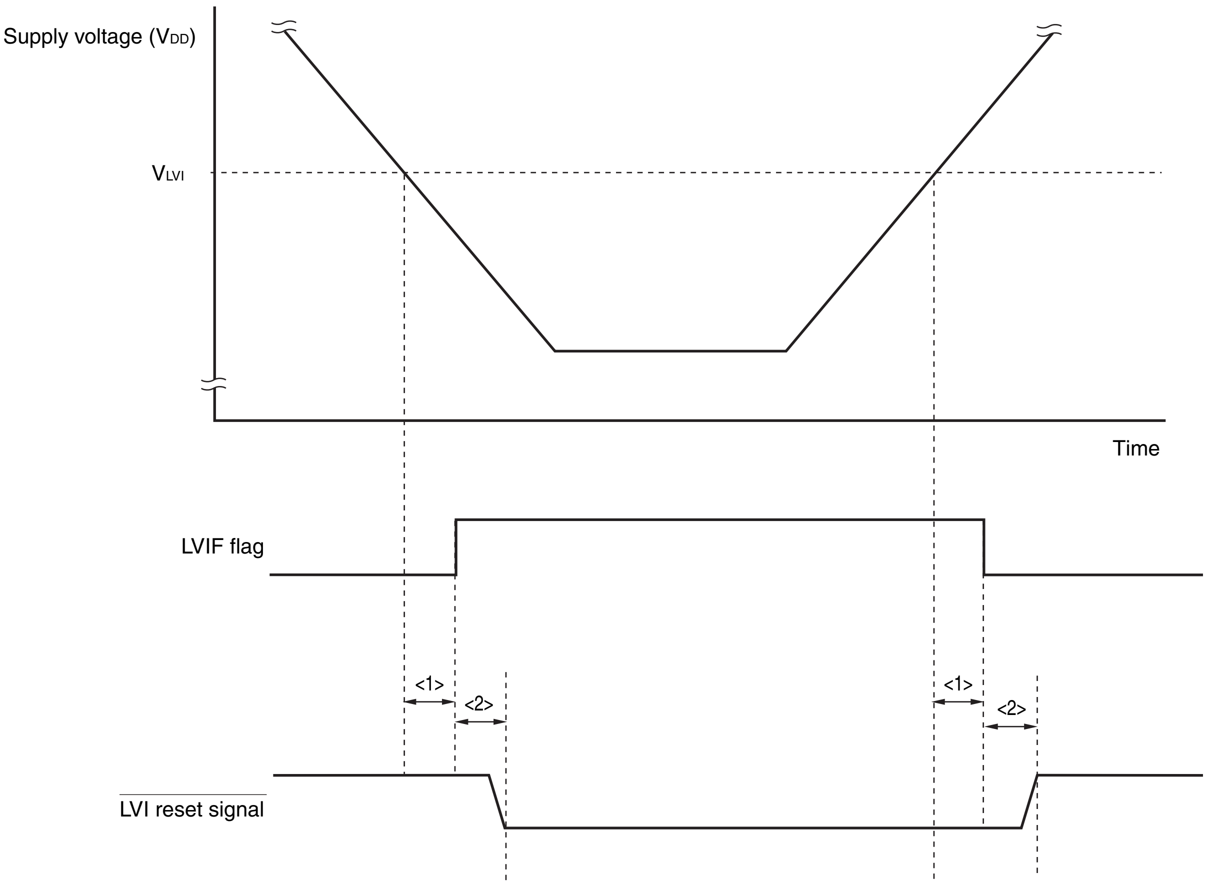
<R> **(2) Delay from the time LVI reset source is generated until the time LVI reset has been generated or released**

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 21-15**).

See the timing in **Figure 20-2 (2) When LVI is ON upon power application (option byte: LVIOFF = 0)** for the reset processing time until the normal operation is entered after the LVI reset is released.

Figure 21-15. 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.))

<2>: Detection delay time (200 μs (MAX.))

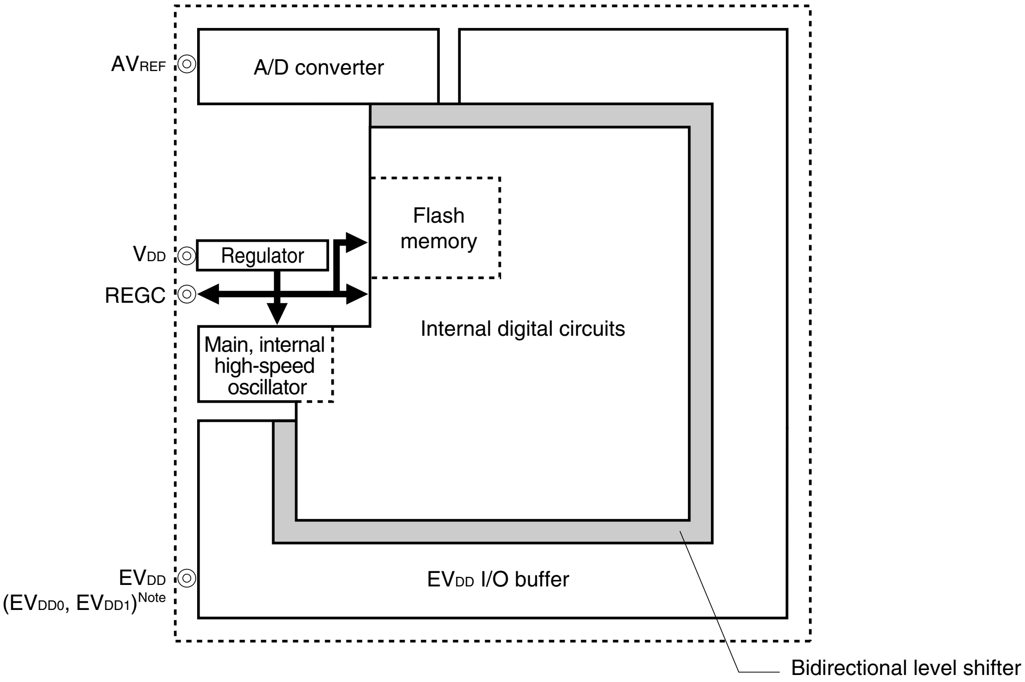
CHAPTER 22 REGULATOR

22.1 Regulator Overview

The 78K0R/Hx3 contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the <R> 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 2.5 V (TYP.).

Figure 22-1. Regulator


AVREF
A/D converter
Flash memory
VDD
Regulator
REGC
Main, internal high-speed oscillator
Internal digital circuits
EVDD
(EVDD0, EVDD1)Note
EVDD I/O buffer
Bidirectional level shifter

Note 78K0R/HG3 only

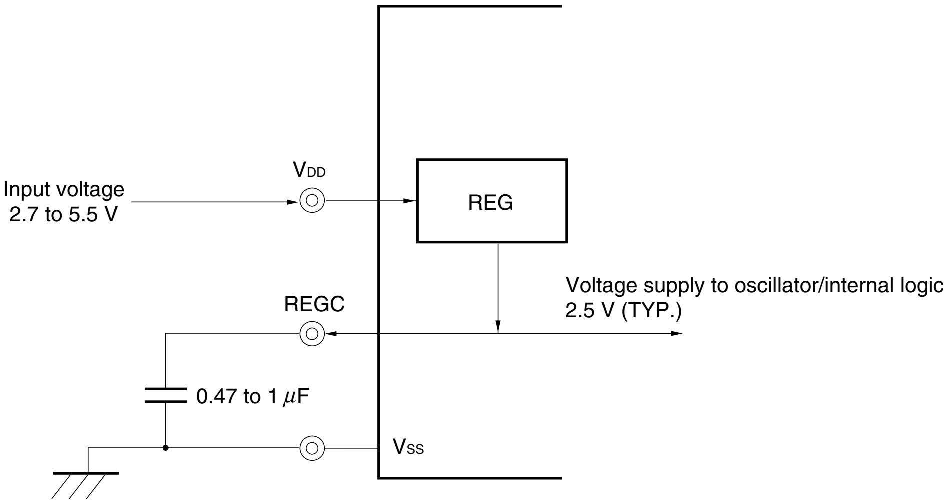
<R> 22.2 Operation

The regulator of this product always operates in all mode (normal operation mode, HALT mode, STOP mode, or during reset).

Be sure to connect a capacitor (0.47 to 1 μF) to the REGC pin to stabilize the regulator output.

A diagram of the regulator pin connection method is shown below.

Figure 22-2. REGC Pin Connection



CHAPTER 23 OPTION BYTE

23.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the 78K0R/Hx3 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 020C0H to 020C3H. Therefore, set the same values as 000C0H to 000C3H to 020C0H to 020C3H.

23.1.1 User option byte (000C0H to 000C2H/020C0H to 020C2H)

(1) 000C0H/020C0H

○ 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 020C0H when the boot swap operation is used because 000C0H is replaced by 020C0H.

(2) 000C1H/020C1H

○ 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, clock monitor, illegal memory access, WDT, or illegal instructions).

○ Internal high-speed oscillator: specification of 4 MHz/8 MHz

○ PLL clock multiplication setting: ×6/×8

○ P130 mode setting

Port mode/alternate-function mode

○ Clock monitor operation

- Operates/stops

○ Setting of Internal low-speed oscillation

- Enabling/stopping set to CPU or peripheral hardware clock

○ Internal low-speed oscillation clock operation in STOP mode

- Operates/stops in STOP mode

Caution Set the same value as 000C1H to 020C1H when the boot swap operation is used because 000C1H is replaced by 020C1H.

(3) 000C2H/020C2H

○ Control of clock output

- Enabling/stopping clock output after a reset is released
- <R> Output clock selection after a reset is released

Caution Set the same value as 000C2H to 020C2H when the boot swap operation is used because 000C2H is replaced by 020C2H

23.1.2 On-chip debug option byte (000C3H/020C3H)

○ 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 020C3H when the boot swap operation is used because 000C3H is replaced by 020C3H.

<R> 23.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 23-1. Format of User Option Byte (000C0H/020C0H) (1/2)

Address: 000C0H/020C0H^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	25%
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 (TYP.)
0	0	0	$2^7/f_{IL}$ (4.27 ms)
0	0	1	$2^8/f_{IL}$ (8.53 ms)
0	1	0	$2^9/f_{IL}$ (17.07 ms)
0	1	1	$2^{10}/f_{IL}$ (34.13 ms)
1	0	0	$2^{12}/f_{IL}$ (136.5 ms)
1	0	1	$2^{14}/f_{IL}$ (546.1 ms)
1	1	0	$2^{15}/f_{IL}$ (1092 ms)
1	1	1	$2^{17}/f_{IL}$ (4369 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 020C0H when the boot swap operation is used because 000C0H is replaced by 020C0H.

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.

Remarks 1. f_{IL}: Internal low-speed oscillation clock frequency

2. ( ): f_{IL} = 30 kHz (TYP.)

Figure 23-2. Format of Option Byte (000C1H/020C1H)

Address: 000C1H/020C1H[Note]

7	6	5	4	3	2	1	0
LIOUSE	LIOSTOPB	LIOSYSB	CLKMB	RESOUTB	OPTPLL	SEL4M	LVIOFF

LIOUSE	Internal low-speed oscillation operation control
0	Stops internal low-speed oscillation operation.
1	Enables internal low-speed oscillation operation.

LIOSTOPB	Internal low-speed oscillation setting in STOP mode
0	Stops internal low-speed oscillation in STOP mode.
1	Operates internal low-speed oscillation in STOP mode.

LIOSYSB	Enabling/disabling setting internal low-speed oscillation (f_{IL}) to CPU/peripheral hardware clock (f_{CLK})
0	Enables internal low-speed oscillation (f_{IL}) operation setting for system clock (f_{CLK}).
1	Disables internal low-speed oscillation (f_{IL}) operation setting for system clock (f_{CLK}).

CLKMB	Clock monitoring operation control
0	Operates clock monitoring.
1	Stops clock monitoring.

RESOUTB	P130 function selection
0	Uses P130 as RESOUT pin. • Outputs low level during RESET. • Automatically outputs high level when RESET released. • Port latch value does not affect output.
1	Uses P130 as normal port (dedicated to output). • Outputs low level during RESET. • Outputs port latch after RESET release.

OPTPLL	PLL multiplication selection
0	×8 multiplication
1	×6 multiplication

SEL4M	Internal high-speed oscillator frequency division selection
0	8 MHz operation
1	4 MHz operation

LVIOFF	Setting of LVI on power application
0	LVI is ON by default (LVI default start function enabled) after reset release (upon power application)
1	LVI is OFF by default (LVI default start function stopped) after reset release (upon power application)

Note Set the same value as 000C1H to 020C1H when the boot swap operation is used because 000C1H is replaced by 020C1H.

(**Caution** is given on the next page.)

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, illegal instruction execution, detection of a main clock oscillation stop by clock moniter, and detection of illegal memory access occur.**

Figure 23-3. Format of Option Byte (000C2H/020C2H)

Address: 000C2H/020C2H[Note 1]

7	6	5	4	3	2	1	0
PCLOFF	1	1	CSEL1B	CSEL0B	CCS2B	CCS1B	CCS0B

PCLOFF	Enabling/disabling default PCL output
0	PCL default output enable
1	PCL default output disable

CSEL1B	CSEL0B	CCS2B	CCS1B	CCS0B	PCL output clock selection[Note 2]				
						f_{MAIN} = 8 MHz	f_{PLL} = 16 MHz	f_{MAIN} = 20 MHz	f_{PLL} = 24 MHz
1	1	1	1	1	f_{MAIN}	8MHz	–	Setting prohibited[Note3]	–
1	1	1	1	0	$f_{MAIN}/2$	4 MHz	–	10 MHz	–
1	1	1	0	1	$f_{MAIN}/2^2$	2 MHz	–	5 MHz	–
1	1	1	0	0	$f_{MAIN}/2^3$	1 MHz	–	2.5 MHz	–
1	1	0	1	1	$f_{MAIN}/2^4$	0.5 MHz	–	1.25 MHz	–
1	1	0	1	0	$f_{MAIN}/2^{11}$	3.91 kHz	–	9.76 kHz	–
1	1	0	0	1	$f_{MAIN}/2^{12}$	1.95 kHz	–	4.88 kHz	–
1	1	0	0	0	$f_{MAIN}/2^{13}$	0.98 kHz	–	2.44 kHz	–
1	0	1	1	1	f_{PLL}	–	Setting prohibited[Note 3]	–	Setting prohibited[Note 3]
1	0	1	1	0	$f_{PLL}/2$	–	8MHz	–	12 MHz
1	0	1	0	1	$f_{PLL}/2^2$	–	4 MHz	–	6 MHz
1	0	1	0	0	$f_{PLL}/2^3$	–	2 MHz	–	3 MHz
1	0	0	1	1	$f_{PLL}/2^4$	–	1 MHz	–	1.5 MHz
1	0	0	1	0	$f_{PLL}/2^{11}$	–	7.81 kHz	–	11.72 kHz
1	0	0	0	1	$f_{PLL}/2^{12}$	–	3.91 kHz	–	5.86 kHz
1	0	0	0	0	$f_{PLL}/2^{13}$	–	1.95 kHz	–	2.93 kHz
0	1	×	×	×	f_{IL}	30 kHz (TYP.)			
0	0	×	×	×	f_{SUB}	See **CHAPTERS 29 ELECTRICAL SPECIFICATIONS**			
Other than above					Setting prohibited				

Notes 1. Set the same value as 000C2H to 020C2H when the boot swap operation is used because 000C2H is replaced by 020C2H

2. When PCLOFF is 0, select either f_{MAIN} or a divided f_{MAIN} as the PCL output clock. Note that the PCL clock is also output in flash memory programming mode.

3. Setting an output clock exceeding 12 MHz is prohibited

(**Cautions 1 to 4** and **Remarks 1 to 3** are given on the next page.)

Cautions 1. Change the output clock after disabling clock output (PCLOE = 0).

2. If the selected clock (f_{MAIN} or f_{PLL} or f_{IL} or f_{SUB}) stops during clock output (PCLOE = 1), the output becomes undefined.

3. Errors may be caused in the PCL output frequency due to fluctuation of EV_{DD} or EV_{SS}. Perform a thorough evaluation when a highly accurate clock is required.

4. Be sure to set bits 6 and 5 to “1”.

Remarks 1.	f_{MAIN}:	Main system clock frequency
2.	f_{PLL}:	PLL clock frequency
3.	f_{IL}:	Internal low-speed oscillation clock frequency
4.	f_{SUB}:	Subclock frequency
5.	×:	Don’t care

23.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 23-4. Format of On-chip Debug Option Byte (000C3H/020C3H)

Address: 000C3H/020C3H[Note]

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Enables on-chip debugging. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debugging. Does not erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Note Set the same value as 000C3H to 020C3H when the boot swap operation is used because 000C3H is replaced by 020C3H.

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.

<R> 23.4 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.

Refer to 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          C9H       ;  Enables internal low-speed oscillation operation
                              ;  Operates internal low-speed oscillation in STOP mode
                              ;  Enables internal low-speed oscillation operation setting for fCLK
                              ;  Operates clock monitoring
                              ;  Uses P130 as dedicated to output port
                              ;  PLL clock multiplication setting: ×8
                              ;  8 MHz is selected as internal high-speed oscillation
                              ;  Stops LVI default start function
        DB          7FH       ;  Enables PCL default output
                              ;  fMAIN is selected as PCL output clock
        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 020C0H to 020C3H. Describe to 020C0H to 020C3H, therefore, the same values as 000C0H to 000C3H as follows.

```
OPT2  CSEG   AT   020C0H
      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          C9H      ; Enables internal low-speed oscillation operation
                           ; Operates internal low-speed oscillation in STOP mode
                           ; Enables internal low-speed oscillation operation setting for fCLK
                           ; Operates clock monitoring
                           ; Uses P130 as dedicated to output port
                           ; PLL clock multiplication setting: ×8
                           ; 8 MHz is selected as internal high-speed oscillation
                           ; Stops  LVI default start function stopped
      DB          7FH      ; Enables PCL default output
                           ; fMAIN is selected as PCL output clock
      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 020C0H to 020C3H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.

CHAPTER 24 FLASH MEMORY

The 78K0R/Hx3 incorporates the flash memory.

78K0R/Hx3 Microcontrollers		Products	Code Flash	Data Flash
78K0R/HC3	48 pins	μPD78F1031	64 KB	16 KB
		μPD78F1032	96 KB	16 KB
		μPD78F1033	128 KB	16 KB
		μPD78F1034	192 KB	16 KB
		μPD78F1035	256 KB	16 KB
78K0R/HE3	64 pins	μPD78F1036	64 KB	16 KB
		μPD78F1037	96 KB	16 KB
		μPD78F1038	128 KB	16 KB
		μPD78F1039	192 KB	16 KB
		μPD78F1040	256 KB	16 KB
78K0R/HF3	80 pins	μPD78F1041	64 KB	16 KB
		μPD78F1042	96 KB	16 KB
		μPD78F1043	128 KB	16 KB
		μPD78F1044	192 KB	16 KB
		μPD78F1045	256 KB	16 KB
78K0R/HG3	100 pins	μPD78F1046	64 KB	16 KB
		μPD78F1047	96 KB	16 KB
		μPD78F1048	128 KB	16 KB
		μPD78F1049	192 KB	16 KB
		μPD78F1050	256 KB	16 KB

24.1 Overview

The code flash memory store program code and constant data.

The data flash is allocated as parts of external memory area. Data flash memory is possible to programming during program operation by Renesas Electronics library (to be supported).

Flash memory is commonly used in the following development environments and applications.

- For altering software after solder-mounting of the microcontroller on the target system.
- For differentiating software in small-scale production of various models.
- For data adjustment according to the user specification when starting mass production.
- For facilitating inventory management.
- For updating software after shipment.

The code flash memory can be written in different ways.

- Mounted on the dedicated Adapter (FA series), and rewriting by communication with dedicated flash memory programmer via serial interface (off-board programming)
- Mounted on the target board, and rewriting by communication with dedicated flash memory programmer via serial interface (on-board programming)
- Rewriting code flash memory by user program (application) (self programming)

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Additionally, code flash memory has option byte area to set option function of 78K0R/Hx3.

The option byte is stop enable/disabled setting by the software of internal low-speed oscillator and setting of operation mode of watchdog timer, etc.

Be sure to set data for option byte area with writing programs to the code flash memory. For details, refer to **CHAPTER 23 OPTION BYTE**.

24.1.1 Code flash memory features

- All-blocks or multiple blocks batch erase or single block erase
- Erase/write with single power supply
- Communication with dedicated flash memory programmer via various serial interfaces
- On-board and off-board programming
- Flash memory programming by self-programming
- Flash memory write prohibit function supported (security function)
- Safe rewriting of entire flash memory area by self programming using boot swap function and flash shield window function
- Interrupts can be acknowledged during self programming

24.1.2 Data flash memory features

The 78K0R/Hx3 products are provided a 16 KB data flash. The data flash is allocated as parts of external memory bus. The data flash has the following features.

- Data flash memory consists of eight blocks (each block size is 2 KB)
- <R> Write access in 32-bit units
- Erase in block units (2 KB)
- Write, erase operations to the data flash memory while application code can be executed (fetches code flash memory)

Caution However, it is not possible to perform an access to the data flash while the code flash is rewritten by self-programming and vice versa.

Remarks 1. Instructions cannot be fetched to the data flash memory area.

2. The data flash area cannot be written by using a dedicated flash memory programmer.

24.2 Registers that Control Flash Memory

(1) Peripheral enable registers 1 (PER1)

These registers are 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.

PER1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears theses registers to 00H.

Caution Set the PER1 register before reading data flash.

Figure 24-1. Format of Peripheral Enable Registers 1 (PER1)

Address: F00F1H After reset: 00H R/W

Symbol	7	6	5	4	<3>	2	<1>	<0>
PER1	0	0	0	0	SAU2EN	0	WUTEN	DFLEN

DFLEN	Control of data flash input clock
0	Stops input clock supply. • Cannot be read and written for data flash area.
1	Supplies input clock. • Can be read and written for data flash area.

WUTEN	Supply of synchronous clock to wakeup timer
0	Stops synchronous clock.
1	Supplies synchronous clock.

SAU2EN	Supply of clock to serial array unit2
0	Stops clock.
1	Supplies clock.

Caution Be sure to clear the following bits to 0.
78K0R/HC3: Bits 2 to 7
78K0R/HE3, 78K0R/HF3, 78K0R/HG3: Bits 2, 4 to 7

(2) 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.

Caution This is not required to be set during data flash programming.

Figure 24-2. Format of Background Event Control Register (BECTL)

Address: FFFBEH After reset: 00H R/W

<R>

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

(3) Data flash status register (DFLST)

Figure 24-3. Format of Data Flash Status Register (DFLST)

Address: F04F0H After reset: 00H R

Symbol	7	6	5	4	3	2	1	<0>
DFLST	0	0	0	0	0	0	0	DFRDY

DFRDY	Data flash setup completion flag
0	Setup is not complete (the data flash memory cannot be accessed).
1	Setup is complete (the data flash memory can be accessed).

24.3 Data Flash Access Procedure

The data flash is stopped after a reset is released and cannot be accessed (read or programmed) as it is.

24.3.1 Data flash read procedure

<1> To read the data flash while the CPU is operating at a frequency exceeding 20 MHz, write 1 to the DMSTP bit (bit 2 of the OSMC register).

<2> Write 1 to the DFLEN bit (bit 0 of the PER1 register). Check the settings of the data flash status register.

<R> <3> Perform a setup wait (50 μs to 74 μs). Check the settings of the data flash status register.
Accessing the data flash before the setup completion is prohibited.
To execute the STOP instruction/HALT instruction before setup completion, set DFLEN = 0 before doing so.

<4> The data flash can be read after setup completion.

24.3.2 Data flash programming procedure

Data flash programming will be supported by using an Renesas Electronics library that corresponds to the data flash while executing an application.

See the manual related to the data flash memory access library (being prepared) for writing using the library.

24.3.3 Data flash stop procedure

The current consumption of the data flash can be suppressed while not using the data flash, by setting the DFLEN bit (bit 0 of the PER1 register) to 0.

After executing the HALT instruction, the flash memory automatically transitions to the HALT mode and the current consumption is suppressed. A setup wait is not required when releasing the HALT mode.

The flash memory automatically transitions to the STOP mode and the current consumption of the data flash can be suppressed to 0 even if the DFLEN bit is not manipulated when the STOP instruction is executed. Check the settings of the data flash status register after the STOP mode is released. The data flash cannot be accessed immediately after the STOP mode is released, because a setup wait is required after the STOP mode is released. When executing the STOP instruction before the data flash setup is completed, set DFLEN = 0 before doing so.

Remark Do not execute a STOP or HALT instruction while the data flash memory is being written or erased.

24.4 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 code flash memory can be rewritten after the 78K0R/Hx3 has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the 78K0R/Hx3 is mounted on the target system.

Table 24-1. Wiring Between 78K0R/HC3 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer			Pin Name	Pin No.
Signal Name	I/O	Pin Function		
SI/RxD[Notes 1, 2]	Input	Receive signal	TOOL0/P40	3
SO/TxD[Note 2]	Output	Transmit signal		
SCK	Output	Transfer clock	–	–
CLK	Output	Clock output	–	–
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	4
FLMD0	Output	Mode signal	FLMD0	7
V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	12
			EV_{DD}	12
			AV_{REF}	35
GND	–	Ground	V_{SS}	11
			EV_{SS}	11
			AV_{SS}	36

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.

Table 24-2. Wiring Between 78K0R/HE3 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer			Pin Name	Pin No.
Signal Name	I/O	Pin Function		
SI/RxD[Notes 1, 2]	Input	Receive signal	TOOL0/P40	5
SO/TxD[Note 2]	Output	Transmit signal		
SCK	Output	Transfer clock	–	–
CLK	Output	Clock output	–	–
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	6
FLMD0	Output	Mode signal	FLMD0	9
V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	15
			EV_{DD}	16
			AV_{REF}	47
GND	–	Ground	V_{SS}	13
			EV_{SS}	14
			AV_{SS}	48

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.

Table 24-3. Wiring Between 78K0R/HF3 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer			Pin Name	Pin No.
Signal Name	I/O	Pin Function		
SI/RxD[Notes 1, 2]	Input	Receive signal	TOOL0/P40	9
SO/TxD[Note 2]	Output	Transmit signal		
SCK	Output	Transfer clock	–	–
CLK	Output	Clock output	–	–
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	10
FLMD0	Output	Mode signal	FLMD0	13
V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	19
			EV_{DD}	20
			AV_{REF}	59
GND	–	Ground	V_{SS}	17
			EV_{SS}	18
			AV_{SS}	60

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.

Table 24-4. Wiring Between 78K0R/HG3 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer			Pin Name	Pin No.
Signal Name	I/O	Pin Function		
SI/RxD[Notes 1, 2]	Input	Receive signal	TOOL0/P40	12
SO/TxD[Note 2]	Output	Transmit signal		
SCK	Output	Transfer clock	–	–
CLK	Output	Clock output	–	–
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	13
FLMD0	Output	Mode signal	FLMD0	16
V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	22
			EV_{DD0}	23
			EV_{DD1}	53
			AV_{REF}	73
GND	–	Ground	V_{SS}	20
			EV_{SS0}	21
			EV_{SS1}	43
			AV_{SS}	74

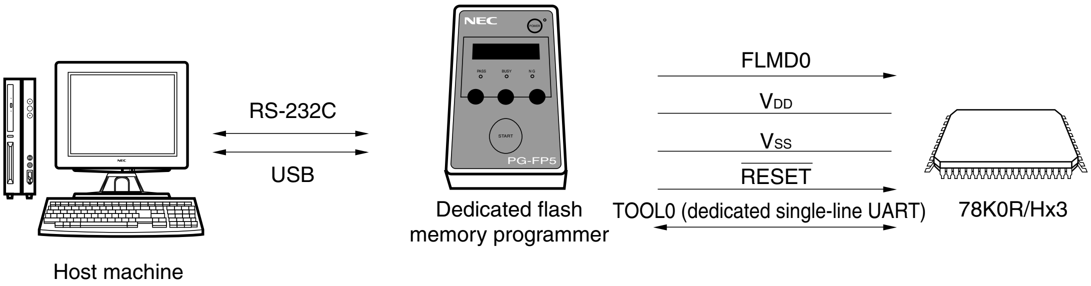
Notes **1.** This pin is not required to be connected when using PG-FP5 or FL-PR5.

2. Connect SI/RxD or SO/TxD when using QB-MINI2.

24.5 Programming Environment

The environment required for writing a program to the code flash memory of the 78K0R/Hx3 is illustrated below.

Figure 24-4. Environment for Writing Program to Code Flash Memory


RS-232C
USB
Host machine
NEC
PG-FP5
Dedicated flash memory programmer
FLMD0
V_{DD}
V_{SS}
$\overline{\text{RESET}}$
TOOL0 (dedicated single-line UART)
78K0R/Hx3

A host machine that controls the dedicated flash memory programmer is necessary.

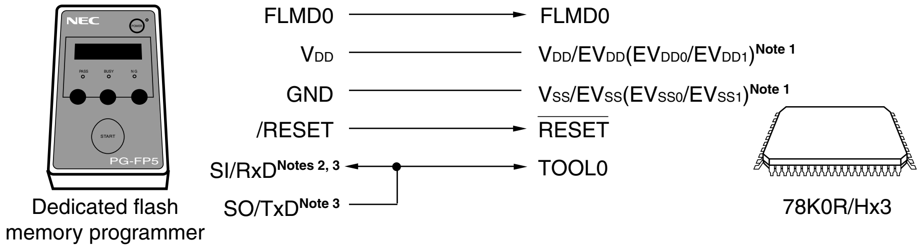
To interface between the dedicated flash memory programmer and the 78K0R/Hx3, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

24.6 Communication Mode

Communication between the dedicated flash memory programmer and the 78K0R/Hx3 is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the 78K0R/Hx3.

Transfer rate: 115,200 bps, 250,000 bps, 500,000, 1M bps

Figure 24-5. Communication with Dedicated Flash Memory Programmer


NEC
PG-FP5
Dedicated flash memory programmer
FLMD0
V_{DD}
GND
/RESET
SI/RxD Notes 2, 3
SO/TxD Note 3
FLMD0
$V_{DD}/EV_{DD}(EV_{DD0}/EV_{DD1})$ Note 1
$V_{SS}/EV_{SS}(EV_{SS0}/EV_{SS1})$ Note 1
$\overline{\text{RESET}}$
TOOL0
78K0R/Hx3

Notes 1. EV_{DD0}, EV_{DD1}, EV_{SS0}, EV_{SS1} are 78K0R/HG3 only

2. This pin is not required to be connected when using PG-FP5 or FL-PR5.

3. Connect SI/RxD or SO/TxD when using QB-MINI2.

When using the dedicated flash memory programmer, the dedicated flash memory programmer generates the following signals for the 78K0R/Hx3. For details, refer to the user's manual for the dedicated flash memory programmer.

Table 24-5. Pin Connection

Dedicated Flash Memory Programmer			78K0R/Hx3	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}, EV_{DD} (EV_{DD0}, EV_{DD1})[Note 1], AV_{REF}	◎
GND	–	Ground	V_{SS}, EV_{SS} (EV_{SS0}, EV_{SS1})[Note 1], AV_{SS}	◎
CLK	Output	Clock output	–	×
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎
SI/RxD[Notes 2, 3]	Input	Receive signal	TOOL0	◎
SO/TxD[Note 3]	Output	Transmit signal		
SCK	Output	Transfer clock	–	×

Notes 1. EV_{DD0}, EV_{DD1}, EV_{SS0}, EV_{SS1} are 78K0R/HG3 only

2. This pin is not required to be connected when using PG-FP5 or FL-PR5.

3. Connect SI/RxD or SO/TxD when using QB-MINI2.

Remark ◎: Be sure to connect the pin.
×: The pin does not have to be connected.

24.7 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.

<R> 24.7.1 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 100 kΩ or more.

(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 **24.2 (2) Back ground event control register**). To pull it down externally, use a resistor of 100 kΩ or more.

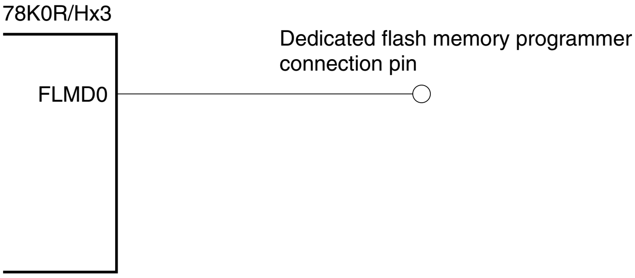
Self programming and the rewriting of code flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V_{SS} pin.

(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 or more.

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

Figure 24-6. FLMD0 Pin Connection Example



<R> 24.7.2 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 EV_{DD} (EV_{DD0}, EV_{DD1})Note via an external resistor.

When on-chip debugging is enabled in the normal operation mode, pull this pin up by connecting it to EV_{DD} (EV_{DD0}, EV_{DD1})Note 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).

Note EV_{DD0}, EV_{DD1} is 78K0R/HG3 only

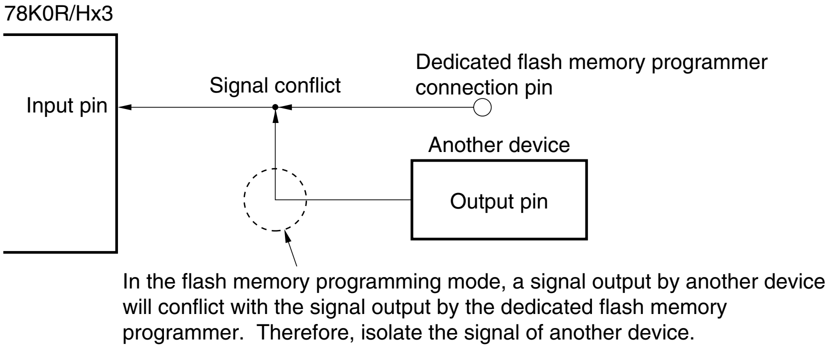
Remark The SAU pins are not used for communication between the 78K0R/Hx3 and dedicated flash memory programmer, because single-line UART is used.

24.7.3 $\overline{\text{RESET}}$ 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 24-7. Signal Conflict ($\overline{\text{RESET}}$ Pin)



In the flash memory programming mode, a signal output by another device will conflict with the signal output by the dedicated flash memory programmer. Therefore, isolate the signal of another device.

24.7.4 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.

<R> 24.7.5 REGC pin

Connect the REGC pin to GND via a capacitor (0.47 to 1 μF) in the same manner as during normal operation. However, when using the STOP mode that has been entered since operation of the internal high-speed oscillation clock and external main system clock, 0.47 μF is recommended. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

24.7.6 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.

24.7.7 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 (EV_{DD}, EV_{DD0} Note, EV_{DD1} Note, EV_{SS}, EV_{SS0} Note, EV_{SS1} Note, AV_{REF}, and AV_{SS}) as those in the normal operation mode.

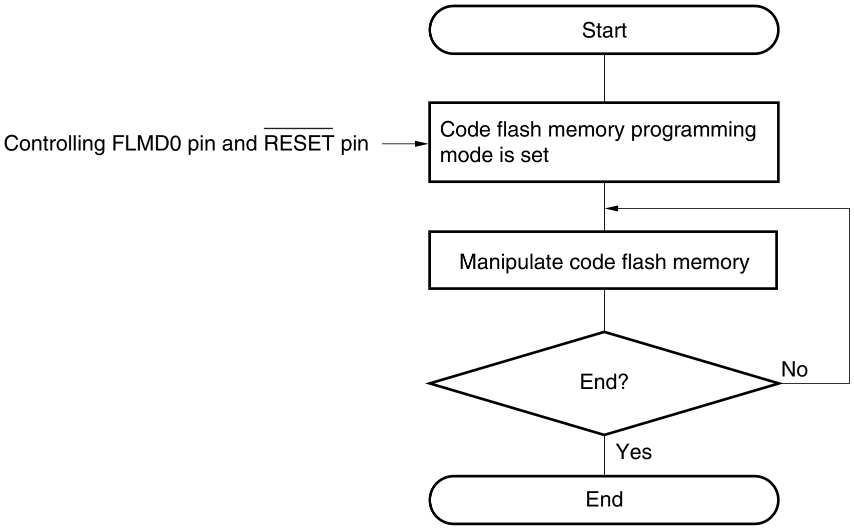
Note EV_{DD0}, EV_{DD1}, EV_{SS0}, EV_{SS1} are 78K0R/HG3 only

24.8 Programming Method

24.8.1 Controlling code flash memory

The following figure illustrates the procedure to manipulate the code flash memory.

Figure 24-8. Code Flash Memory Manipulation Procedure


Start
Controlling FLMD0 pin and RESET pin
Code flash memory programming mode is set
Manipulate code flash memory
End?
No
Yes
End

24.8.2 Code flash memory programming mode

To rewrite the contents of the code flash memory by using the dedicated flash memory programmer, set the 78K0R/Hx3 in the code 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 24-9. Code Flash Memory Programming Mode

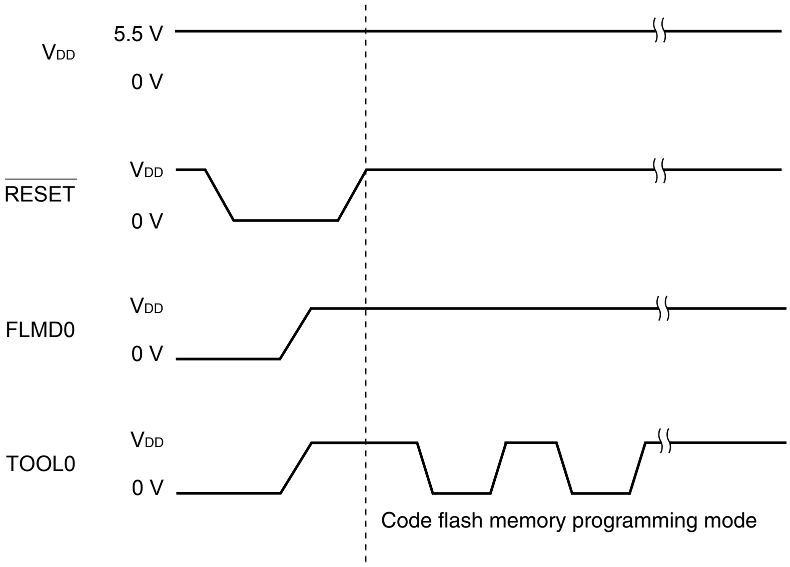


Table 24-6. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0 V	Normal operation mode
V_{DD}	Code flash memory programming mode

24.8.3 Selecting communication mode

Communication mode of the 78K0R/Hx3 as follows.

Table 24-7. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed[Note 2]	Frequency	Multiply Rate	
1-line mode (dedicated single-line UART)	UART-ch0	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOL0

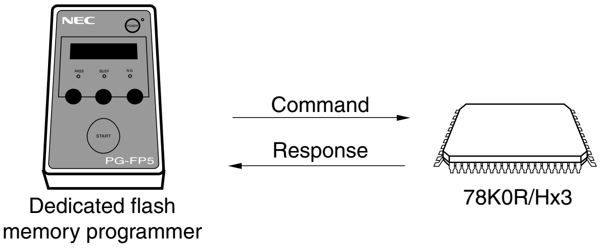
Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

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.

24.8.4 Communication commands

The 78K0R/Hx3 communicates with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0R/Hx3 are called commands, and the signals sent from the 78K0R/Hx3 to the dedicated flash memory programmer are called response.

Figure 24-10. Communication Commands



The flash memory control commands of the 78K0R/Hx3 are listed in the table below. All these commands are issued from the programmer and the 78K0R/Hx3 perform processing corresponding to the respective commands.

Table 24-8. Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the code flash memory with data transmitted from the programmer.
Erase	Chip Erase	Erases the entire flash memory.
	Block Erase	Erases a specified area in the code 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 code flash memory.
Getting information	Silicon Signature	Gets 78K0R/Hx3 information (such as the part number and flash memory configuration).
	Version Get	Gets the 78K0R/Hx3 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/Hx3 return a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0R/Hx3 are listed below.

Table 24-9. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

24.9 Security Settings

The 78K0R/Hx3 supports a security function that prohibits rewriting the user program written to the code 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 code 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 code 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 code 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 code 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
  <R> Execution of the batch erase (chip erase) command, block erase command, and write command on boot cluster 0 (00000H to 01FFFH) in the flash memory is prohibited by this setting.

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 24-10 shows the relationship between the erase and write commands when the 78K0R/Hx3 security function is enabled.

Remark To prohibit writing and erasing during self-programming, use the flash sealed window function (see **24.10.2** for detail).

Table 24-10. 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 **24.10.2** for detail).

Table 24-11. 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		

24.10 Code Flash Memory Programming by Self-Programming

The 78K0R/Hx3 supports a self-programming function that can be used to rewrite the code flash memory via a user program. Because this function allows a user application to rewrite the code flash memory by using the 78K0R/Hx3 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.

Remark For details of the self-programming function and the 78K0R/Hx3 self-programming library, refer to **78K0R Microcontroller Self Programming Library User's Manual (to be prepared)**.

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. Disable DMA operation (DENn = 0) during the execution of self programming library functions.

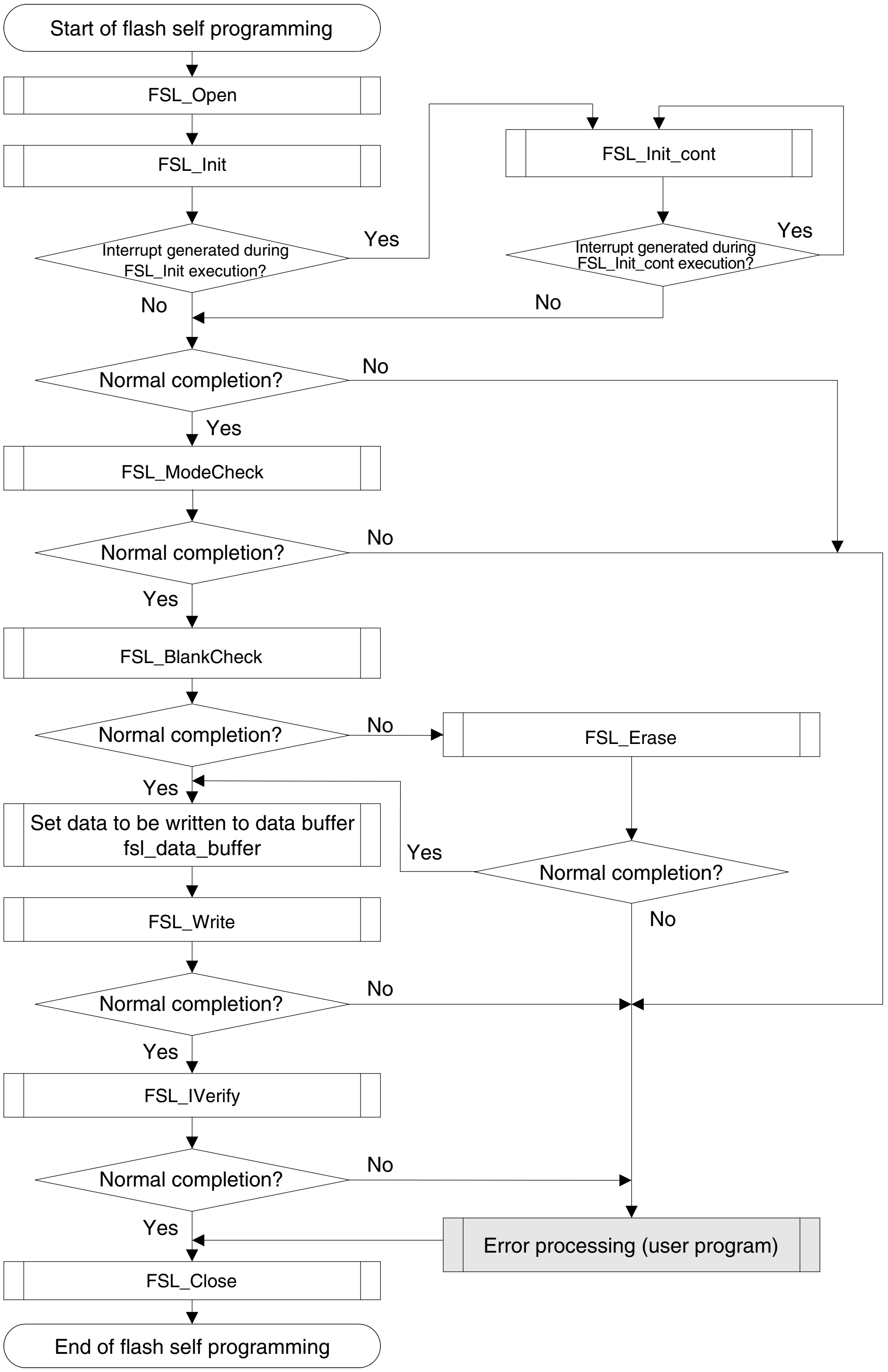
5. While using the self-programming function, the area FFE20H to FFEFFH cannot be used as stack memory.

Remarks 1. For details of the self-programming function and the 78K0R/Hx3 self-programming library, refer to **78K0R Microcontroller Self Programming Library Type2 User's Manual (U19193E).**

2. For details of the time required to execute self programming, see the notes on use that accompany the flash self programming library tool.

The following figure illustrates a flow of rewriting the code flash memory by using a self programming library.

Figure 24-11. Flow of Self Programming (Rewriting Code Flash Memory)



24.10.1 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.

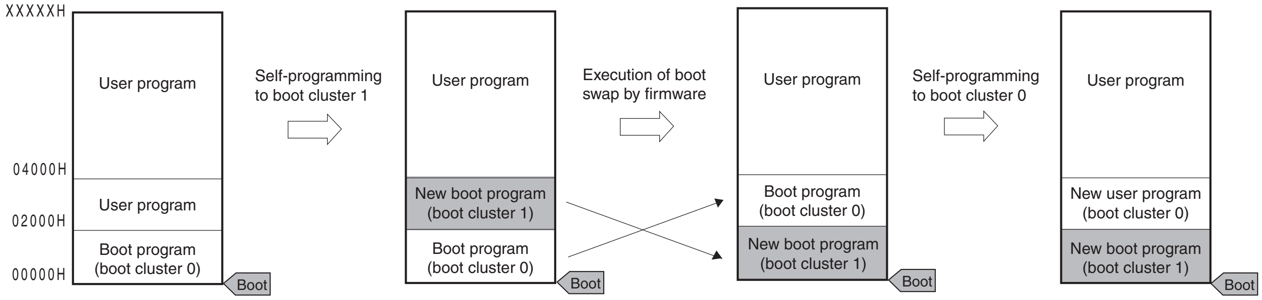
The boot swap function is used to avoid this problem.

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/Hx3, 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 8 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

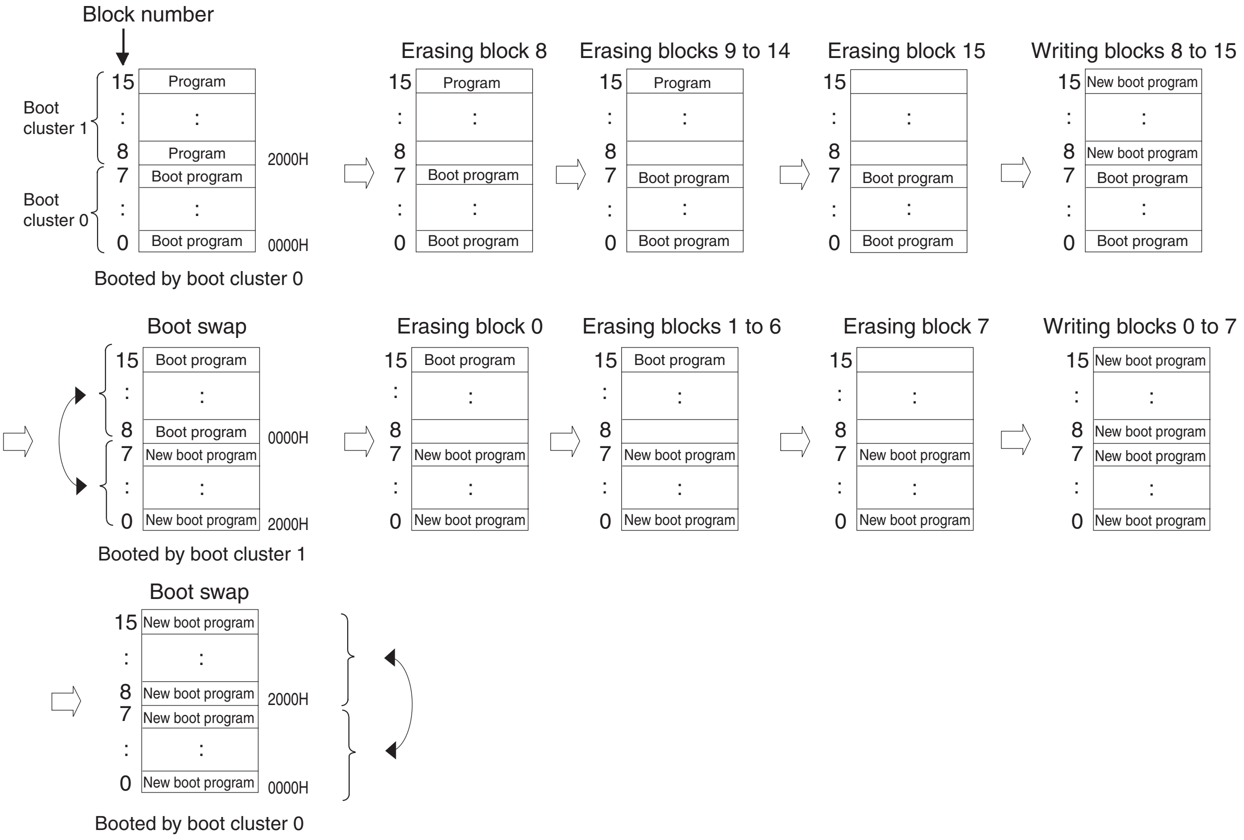
Figure 24-12. Boot Swap Function



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 24-13. Example of Executing Boot Swapping



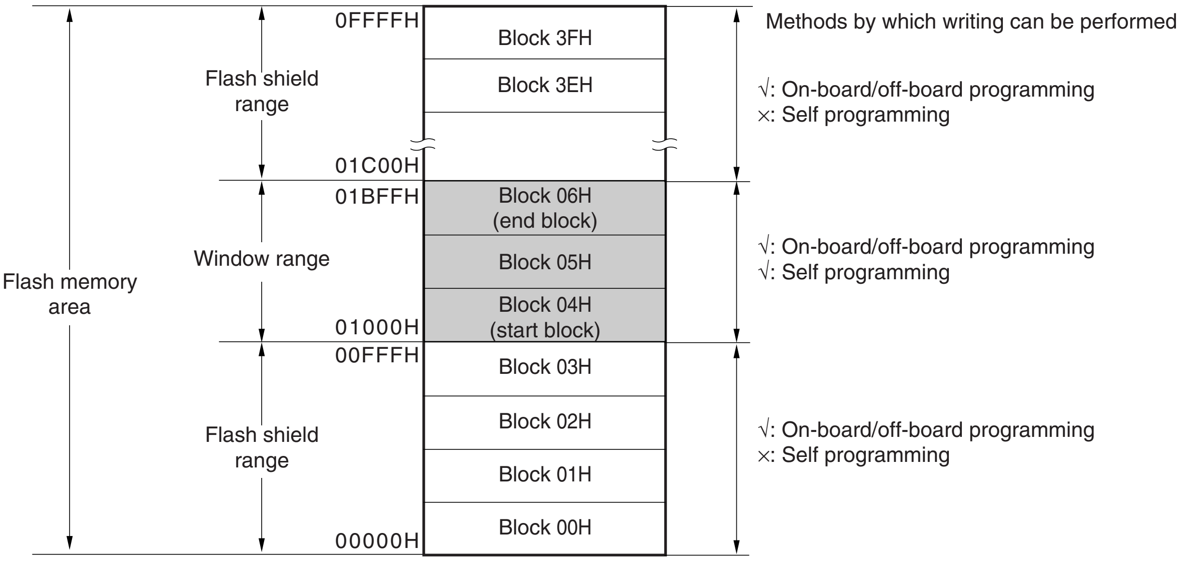
24.10.2 Flash shield window function

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 24-14. Flash Shield Window Setting Example
(Target Device: μPD78F1046, 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 24-12. 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 **24.9 Security Settings** to prohibit writing/erasing during on-board/off-board programming.

24.11 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

http://www2.renesas.com/micro/en/ods/ → Click Version-up Service.

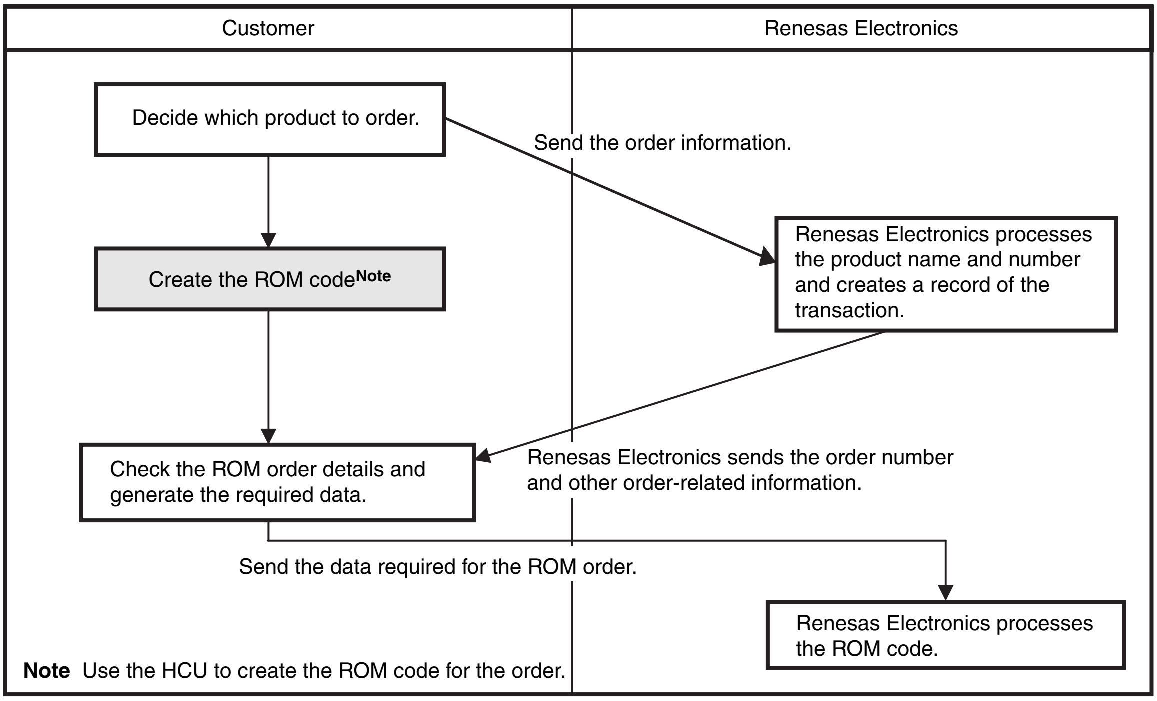
(2) Downloading the HCU

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.

24.11.1 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).


Customer
Renesas Electronics
Decide which product to order.
Send the order information.
Create the ROM codeNote
Renesas Electronics processes the product name and number and creates a record of the transaction.
Check the ROM order details and generate the required data.
Renesas Electronics sends the order number and other order-related information.
Send the data required for the ROM order.
Renesas Electronics processes the ROM code.

Note Use the HCU to create the ROM code for the order.

CHAPTER 25 DATA PROTECTION AND SAFETY

The 78K0R/Hx3 is provided with a safety support function to prevent erroneous operation of the function to protect data in the internal memory from illegal memory access operations as well as erroneous operation of specific registers.

25.1 Data Protection

25.1.1 Illegal-memory access detection function

The 78K0R/Hx3 is provided with an illegal-memory access detection function. The illegal-memory access detection function detects the following access operations and generates an internal reset when an access operation to an illegal area has been detected.

- Write access operation to other than a RAM, an SFR, or a 2nd SFR
- Read access operation to other than a code flash, a RAM, an SFR, a 2nd SFR, or a data flash
- Fetch access operation to other than a code flash or a RAM

(1) Illegal-memory access detection control register (IAWCTL)

The illegal-memory access detection control register is used to select handling of detection of illegal access operations to a memory.

The IAWCTL register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Caution Writing to the IAWCTL register is valid only when bit 6 (GDIAW bit) of the specific-register manipulation protection register (GUARD) is “1”.

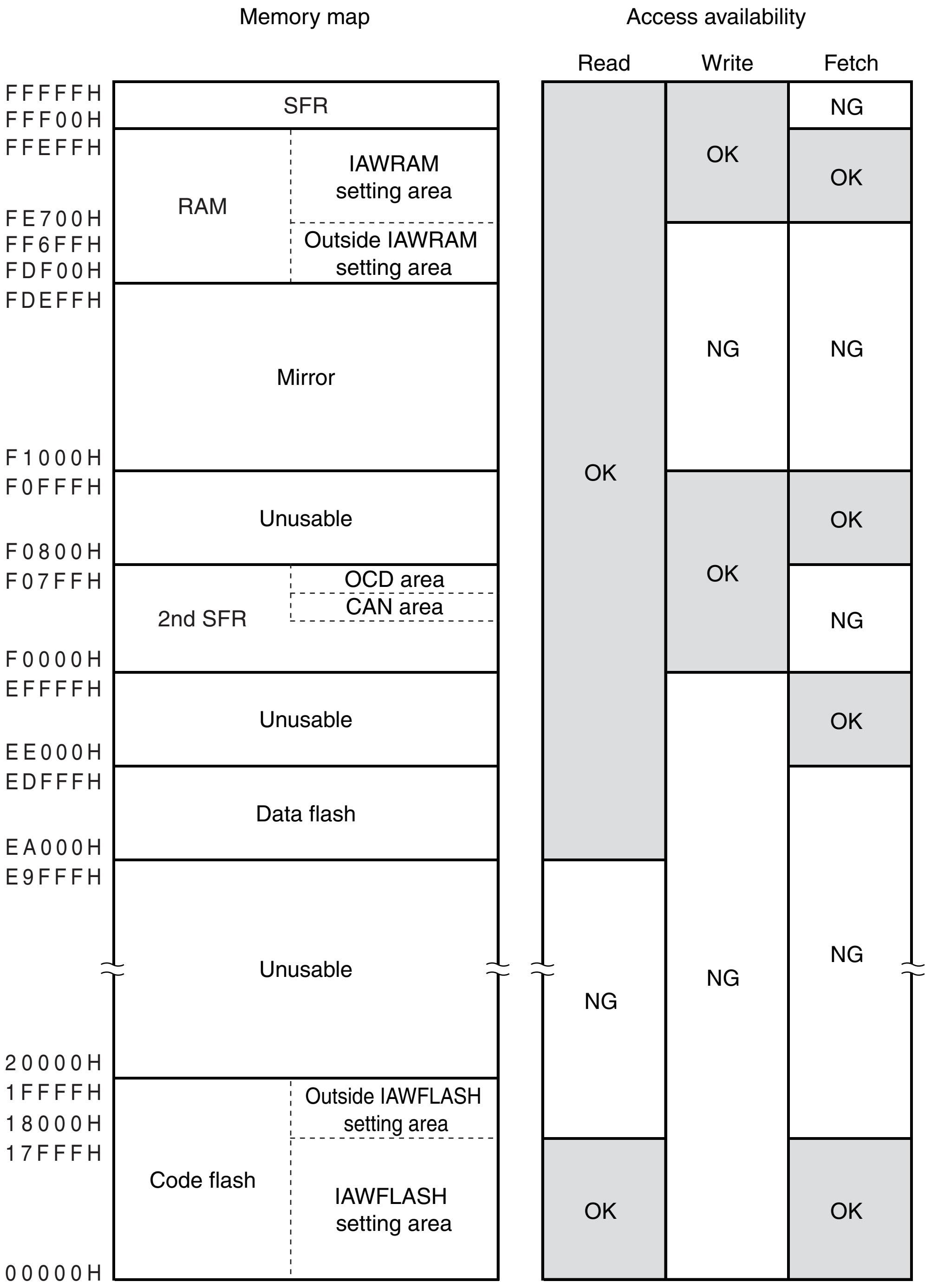
Figure 25-1. Format of Illegal-Memory Access Detection Control Register (IAWCTL)

Address: F0074H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	0	0	0	0	0	0

IAWEN	Illegal-memory access detection function control
0	Invalidates illegal memory access detection.
1	Validates illegal memory access detection.

Figure 25-2. Illegal Memory Access Identification Example
(Setting When Code Flash: 128 KB, IAWFLASH = 0CH, RAM: 8 KB, IAWRAM = 0CH)



Remark A reset is generated when an access-disabled (NG) area is accessed when IAWEN = 1.

(2) Illegal-memory access RAM size setting register (IAWRAM)

The illegal-memory access RAM size setting register is used to set the size of a RAM that is subject to illegal-memory access detection.

The IAWRAM register is set by using an 8-bit memory manipulation.

Reset signal generation sets this register to 00H.

Cautions 1. The IAWRAM register can be written only when bit 6 (GDIAW bit) of the specific-register manipulation protection register (GUARD) is “1”.

2. Rewrite the IAWRAM register after having cleared IAWCTL.IAWEN to “0”.

Figure 25-3. Format of Illegal-Memory Access RAM Size Setting Register (IAWRAM) (1/2)

Address: F0075H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWRAM	0	0	IAWRAM5	IAWRAM4	IAWRAM3	IAWRAM2	IAWRAM1	IAWRAM0

IAWRAM5	IAWRAM4	IAWRAM3	IAWRAM2	IAWRAM1	IAWRAM0	Illegal-memory access detection target RAM size setting
0	0	0	0	0	0	No illegal RAM detection
0	0	0	0	0	1	0.5 KB
0	0	0	0	1	0	1.0 KB
0	0	0	0	1	1	1.5 KB
0	0	0	1	0	0	2 KB
0	0	0	1	0	1	2.5 KB
0	0	0	1	1	0	3 KB
0	0	0	1	1	1	3.5 KB
0	0	1	0	0	0	4 KB
0	0	1	0	0	1	4.5 KB[Note 1]
0	0	1	0	1	0	5 KB[Note 1]
0	0	1	0	1	1	5.5 KB[Note 1]
0	0	1	1	0	0	6 KB[Note 1]
0	0	1	1	0	1	6.5 KB[Note 2]
0	0	1	1	1	0	7 KB[Note 2]
0	0	1	1	1	1	7.5 KB[Note 2]
0	1	0	0	0	0	8 KB[Note 2]

Notes 1. Setting prohibited for products with a RAM of no more than 4 KB

2. Setting prohibited for products with a RAM of no more than 6 KB

Figure 25-3. Format of Illegal-Memory Access RAM Size Setting Register (IAWRAM) (2/2)

IAWRAM5	IAWRAM4	IAWRAM3	IAWRAM2	IAWRAM1	IAWRAM0	Illegal-memory access detection target RAM size setting
0	1	0	0	0	1	8.5 KB[Note 1]
0	1	0	0	1	0	9 KB[Note 1]
0	1	0	0	1	1	9.5 KB[Note 1]
0	1	0	1	0	0	10 KB[Note 1]
0	1	0	1	0	1	10.5 KB[Note 1]
0	1	0	1	1	0	11 KB[Note 1]
0	1	0	1	1	1	11.5 KB[Note 1]
0	1	1	0	0	0	12 KB[Note 1]
0	1	1	0	0	1	12.5 KB[Note 2]
0	1	1	0	1	0	13 KB[Note 2]
0	1	1	0	1	1	13.5 KB[Note 2]
0	1	1	1	0	0	14 KB[Note 2]
0	1	1	1	0	1	14.5 KB[Note 2]
0	1	1	1	1	0	15 KB[Note 2]
0	1	1	1	1	1	15.5 KB[Note 2]
1	0	0	0	0	0	16 KB[Note 2]
Other than above						Setting prohibited

Notes 1. Setting prohibited for products with a RAM of no more than 8 KB

2. Setting prohibited for products with a RAM of no more than 12 KB

(3) Illegal-memory access FLASH size setting register (IAWFLASH)

The illegal-memory access FLASH size setting register is used to set the size of a code flash memory that is subject to illegal-memory access detection.

The IAWFLASH register is set by using an 8-bit memory manipulation.

Reset signal generation sets this register to 00H.

Cautions 1. The IAWFLASH register can be written only when bit 6 (GDIAW bit) of the specific-register manipulation protection register (GUARD) is "1".

2. Rewrite the IAWFLASH register after having cleared IAWCTL.IAWEN to "0".

3. If an instruction is executed for an address in the last 16 bytes of the memory set by using the IAWFLASH register, a reset will occur. Therefore, do not allocate code to this area.

Figure 25-4. Format of Illegal-Memory Access FLASH Size Setting Register (IAWFLASH) (1/2)

Address: F0076H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWFLASH	0	0	IAWFL5	IAWFL4	IAWFL3	IAWFL2	IAWFL1	IAWFL0

IAWFL5	IAWFL4	IAWFL3	IAWFL2	IAWFL1	IAWFL0	Illegal-memory access detection target flash memory size setting
0	0	0	0	0	0	No illegal code flash detection
0	0	0	0	0	1	8 KB
0	0	0	0	1	0	16 KB
0	0	0	0	1	1	24 KB
0	0	0	1	0	0	32 KB[Note 1]
0	0	0	1	0	1	40 KB[Note 2]
0	0	0	1	1	0	48 KB[Note 2]
0	0	0	1	1	1	56 KB[Note 3]
0	0	1	0	0	0	64 KB[Note 3]
0	0	1	0	0	1	72 KB[Note 4]
0	0	1	0	1	0	80 KB[Note 4]
0	0	1	0	1	1	88 KB[Note 4]
0	0	1	1	0	0	96 KB[Note 4]
0	0	1	1	0	1	104 KB[Note 5]
0	0	1	1	1	0	112 KB[Note 5]
0	0	1	1	1	1	120 KB[Note 5]
0	1	0	0	0	0	128 KB[Note 5]

Notes 1. Setting prohibited for products with a code flash memory of no more than 24 KB

2. Setting prohibited for products with a code flash memory of no more than 32 KB

3. Setting prohibited for products with a code flash memory of no more than 48 KB

4. Setting prohibited for products with a code flash memory of no more than 64 KB

5. Setting prohibited for products with a code flash memory of no more than 96 KB

Figure 25-4. Format of Illegal-Memory Access FLASH Size Setting Register (IAWFLASH) (2/2)

IAWFL5	IAWFL4	IAWFL3	IAWFL2	IAWFL1	IAWFL0	Illegal-memory access detection target flash memory size setting
0	1	0	0	0	1	136 KB[Note 1]
0	1	0	0	1	0	144 KB[Note 1]
0	1	0	0	1	1	152 KB[Note 1]
0	1	0	1	0	0	160 KB[Note 1]
0	1	0	1	0	1	168 KB[Note 1]
0	1	0	1	1	0	176 KB[Note 1]
0	1	0	1	1	1	184 KB[Note 1]
0	1	1	0	0	0	192 KB[Note 1]
0	1	1	0	0	1	200 KB[Note 2]
0	1	1	0	1	0	208 KB[Note 2]
0	1	1	0	1	1	216 KB[Note 2]
0	1	1	1	0	0	224 KB[Note 2]
0	1	1	1	0	1	232 KB[Note 2]
0	1	1	1	1	0	240 KB[Note 2]
0	1	1	1	1	1	248 KB[Note 2]
1	0	0	0	0	0	256 KB[Note 2]
Other than above						Setting prohibited

Notes **1.** Setting prohibited for products with a code flash memory of no more than 128 KB

2. Setting prohibited for products with a code flash memory of no more than 192 KB

25.2 Safety Support Function

The 78K0R/Hx3 is provided with a safety support function to prevent generation of reset signals due to erroneous operation of specific register.

(1) Specific-register manipulation protection register (GUARD)

The GUARD register controls specific registers of the low-voltage detector (LVI) and for clock control, such that writing to them is valid only when manipulating them has been enabled.

Writing to specific registers is ignored when manipulating them has not been enabled by using the GUARD register. Writing from a user program is ignored, but writing to a status flag that is accessed from a peripheral function is not ignored.

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

Reset signal generation sets this register to 00H.

Figure 25-5. Format of Specific-Register Manipulation Protection Register (GUARD) (1/2)

Address: F0070H After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	2	<1>	<0>
GUARD	GDWDT	GDIAW	GDLVI	0	0	0	GDPLL	GDCSC

GDWDT	Protection against manipulation of register used to change the operating mode of the watchdog timer during self programming (WDTSELF)
0	Disables manipulation of WDTSELF register.
1	Enables manipulation of WDTSELF register.
The GDWDT bit is used to select handling of writing the register used to change the operating mode of the watchdog timer during self programming (WDTSELF). Writing the WDTSELF register is valid only when the GDWDT bit is set.	

GDIAW	Protection against manipulation of illegal-memory access detection control register (IAWCTL), illegal-memory access FLASH size setting register (IAWFLASH), and illegal-memory access RAM size setting register (IAWRAM)
0	Disables manipulation of IAWCTL, IAWFLASH, and IAWRAM registers.
1	Enables manipulation of IAWCTL, IAWFLASH, and IAWRAM registers.
The GDIAW bit is used to select handling of writing the illegal-memory access detection control register (IAWCTL), illegal-memory access FLASH size setting register (IAWFLASH), and illegal-memory access RAM size setting register (IAWRAM). Writing the IAWCTL, IAWFLASH, and IAWRAM registers is valid only when the GDIAW bit is set.	

GDLVI	Protection against manipulation of low-voltage detection register (LVIM) and low-voltage detection level selection register (LVIS)
0	Disables manipulation of LVIM and LVIS registers.
1	Enables manipulation of LVIM and LVIS registers.
The GDLVI bit is used to select handling of writing the low-voltage detection register (LVIM) and low-voltage detection level selection register (LVIS). Writing the LVIM and LVIS registers is valid only when the GDLVI bit is set.	

Figure 25-5. Format of Specific-Register Manipulation Protection Register (GUARD) (2/2)

GDPLL	Protection against manipulation of PLL control register (PLLCTL)
0	Disables manipulation of PLLCTL register.
1	Enables manipulation of PLLCTL register.
The GDPLL bit is used to select handling of writing the PLL control register (PLLCTL). Writing the PLLCTL register is valid only when the GDPLL bit is set.	

GDCSC	Protection against manipulation of clock operation status control register (CSC)
0	Disables manipulation of CSC register.
1	Enables manipulation of CSC register.
The GDCSC bit is used to select handling of writing the clock operation status control register (CSC). Writing the CSC register is valid only when the GDCSC bit is set.	

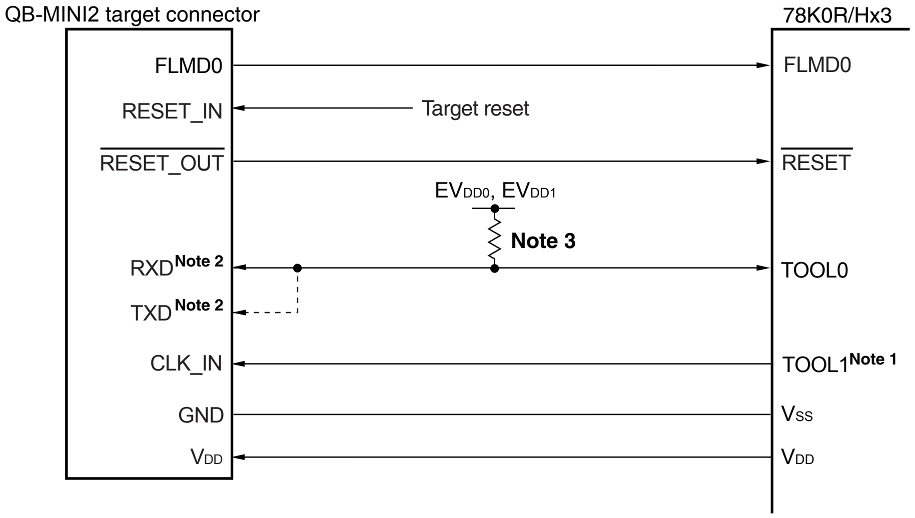
CHAPTER 26 ON-CHIP DEBUG FUNCTION

26.1 Connecting QB-MINI2 to 78K0R/Hx3

The 78K0R/Hx3 uses the V_{DD}, FLMD0, $\overline{\text{RESET}}$, TOOL0, TOOL1[Note], and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2).

Caution The 78K0R/Hx3 has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 26-1. Connection Example of QB-MINI2 and 78K0R/Hx3



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 2-3 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.

3. Recommended pull-up 10 kΩ. In case on-chip debugging is enabled the external pull-up resistor is mandatory to ensure a proper operation of the device when the QB-MINI2 is not connected.

Caution When communicating in 2-line mode, a clock with a frequency of half that of the CPU clock frequency is output from the TOOL1 pin. A resistor or ferrite bead can be used as a countermeasure against fluctuation of the power supply caused by that clock.

Remark The FLMD0 pin is recommended to be open for self-programming. 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 26-1 lists the differences between 1-line mode and 2-line mode.

Table 26-1. Lists the Differences Between 1-line Mode and 2-line Mode.

Communication Mode	Flash Memory Programming Function
1-line mode	Available
2-line mode	None

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.

<R> 26.2 On-Chip Debug Security ID

The 78K0R/Hx3 has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 23 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 020C3H and 020C4H to 020CDH in advance, because 000C3H, 000C4H to 000CDH and 020C3H, and 020C4H to 020CDH 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 26-2. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
020C4H to 020CDH	

<R> 26.3 Securing of User Resources

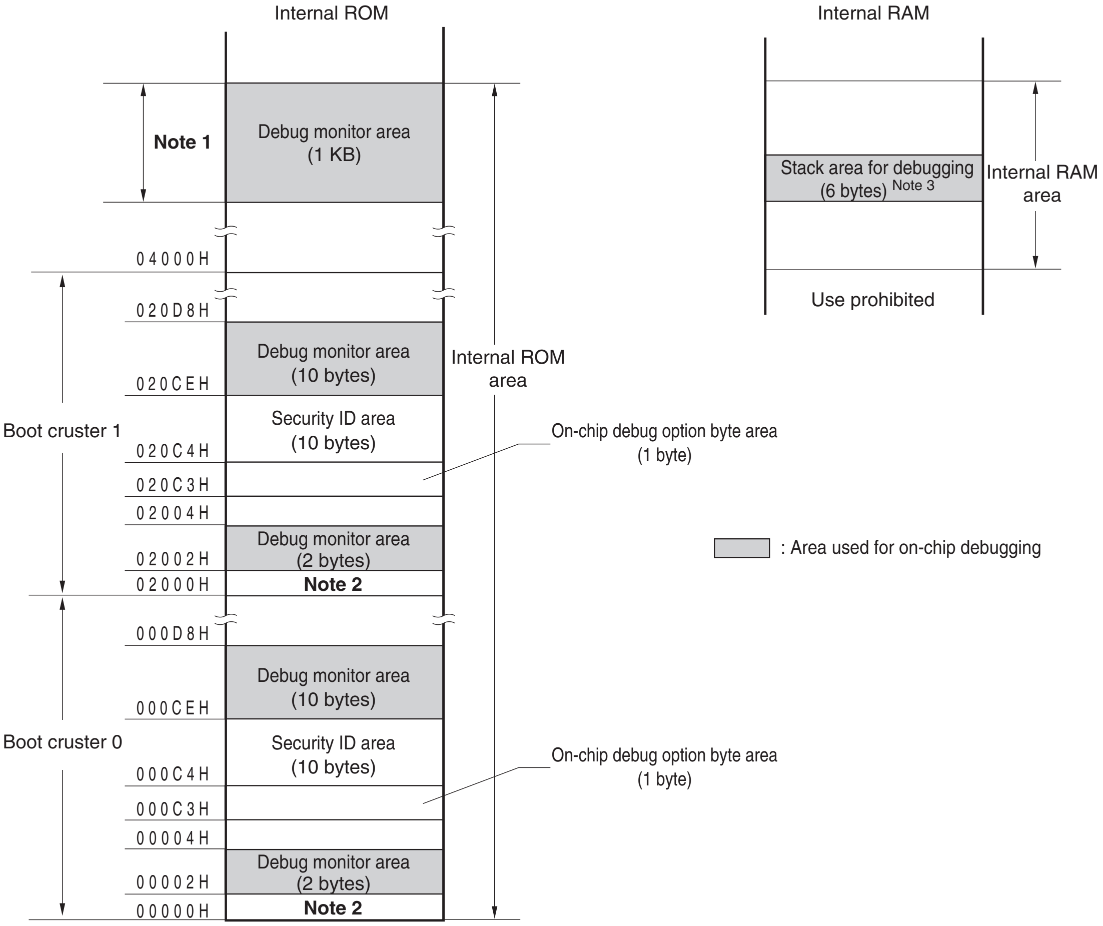
To perform communication between the 78K0R/Hx3 and QB-MINI2, as well as each debug function, the securing of memory space must be done beforehand.

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 26-2 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 26-2. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on products as follows.

Products	Internal ROM	Address
μPD78F1031, 78F1036, 78F1041, 78F1046	64 KB	0FC00H to 0FFFFH
μPD78F1032, 78F1037, 78F1042, 78F1047	96 KB	17C00H to 17FFFH
μPD78F1033, 78F1038, 78F1043, 78F1048	128 KB	1FC00H to 1FFFFH
μPD78F1034, 78F1039, 78F1044, 78F1049	192 KB	2FC00H to 2FFFFH
μPD78F1035, 78F1040, 78F1045, 78F1050	256 KB	3FC00H to 3FFFFH

2. In debugging, reset vector is rewritten to address allocated to a monitor program.

3. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 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 27 BCD CORRECTION CIRCUIT

27.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.

The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/subtracting the BCDADJ register.

27.2 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 27-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

27.3 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 28 INSTRUCTION SET

This chapter lists the instructions in the 78K0R 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 28-5 Operation List** indicate the operation or instruction format that is newly added for the 78K0R microcontrollers.

28.1 Conventions Used in Operation List

28.1.1 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 28-1. Operand Identifiers and Specification Methods

Identifier	Description Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol (SFR symbol) FFF00H to FFFFFH Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only[Note]) FFF00H to FFFFFH
saddr saddrp	FFE20H to FFF1FH Immediate data or labels FFE20H to FF1FH Immediate data or labels (even addresses only[Note])
addr20 addr16 addr5	00000H to FFFFFH Immediate data or labels 0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions[Note]) 0080H to 00BFH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

Remark The special function registers can be described to operand sfr as symbols. See **Table 3-5 SFR List** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Table 3-6 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

28.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 28-2. 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
<R> 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)

28.1.3 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 28-3. 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

28.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

A interrupt and DMA transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 28-4. 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.

<R> 28.2 Operation List

Table 28-5. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	**Note 2**	**Note 3**		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 4]	1	1	–	–	A ← r			
		r, A [Note 4]	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, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash memory area is accessed.

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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (2/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	**Note 2**	**Note 3**		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	4	A ← (ES, addr16)			
		ES:!addr16, A	4	2	–	–	(ES, addr16) ← A			
		A, ES:[DE]	2	2	5	4	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	4	A ← ((ES, DE) + byte)			
		ES:[DE + byte], A	3	2	–	–	((ES, DE) + byte) ← A			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (3/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	Note 2	Note 3		Z	AC	CY
8-bit data transfer	MOV	A, ES:[HL]	2	2	5	4	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	4	A ← ((ES, HL) + byte)			
		ES:[HL + byte], A	3	2	–	–	((ES, HL) + byte) ← A			
		A, ES:[HL + B]	3	2	5	4	A ← ((ES, HL) + B)			
		ES:[HL + B], A	3	2	–	–	((ES, HL) + B) ← A			
		A, ES:[HL + C]	3	2	5	4	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	4	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	4	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	4	A ← ((ES, BC) + word)			
		ES:word[BC], A	4	2	–	–	((ES, BC) + word) ← A			
		B, ES:!addr16	4	2	5	4	B ← (ES, addr16)			
		C, ES:!addr16	4	2	5	4	C ← (ES, addr16)			
		X, ES:!addr16	4	2	5	4	X ← (ES, addr16)			
	XCH	A, r Note 4	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, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash memory area is accessed.

4. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (fCPU) 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (4/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	**Note 2**	**Note 3**		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 4]	1	1	–	–	AX ← rp			
		rp, AX [Note 4]	1	1	–	–	rp ← AX			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash memory area is accessed.

4. 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (5/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	Note 2	Note 3		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	4	AX ← (ES, addr16)			
		ES:!addr16, AX	4	2	–	–	(ES, addr16) ← AX			
		AX, ES:[DE]	2	2	5	4	AX ← (ES, DE)			
		ES:[DE], AX	2	2	–	–	(ES, DE) ← AX			
		AX, ES:[DE + byte]	3	2	5	4	AX ← ((ES, DE) + byte)			
		ES:[DE + byte], AX	3	2	–	–	((ES, DE) + byte) ← AX			
		AX, ES:[HL]	2	2	5	4	AX ← (ES, HL)			
		ES:[HL], AX	2	2	–	–	(ES, HL) ← AX			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (6/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	Note 2	Note 3		Z	AC	CY
16-bit data transfer	MOVW	AX, ES:[HL + byte]	3	2	5	4	AX ← ((ES, HL) + byte)			
		ES:[HL + byte], AX	3	2	–	–	((ES, HL) + byte) ← AX			
		AX, ES:word[B]	4	2	5	4	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	–	–	((ES, B) + word) ← AX			
		AX, ES:word[C]	4	2	5	4	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	–	–	((ES, C) + word) ← AX			
		AX, ES:word[BC]	4	2	5	4	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	–	–	((ES, BC) + word) ← AX			
		BC, ES:!addr16	4	2	5	4	BC ← (ES, addr16)			
		DE, ES:!addr16	4	2	5	4	DE ← (ES, addr16)			
		HL, ES:!addr16	4	2	5	4	HL ← (ES, addr16)			
	XCHW	AX, rp Note 4	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 5	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	4	A, CY ← A + (ES, addr16)	×	×	×
		A, ES:[HL]	2	2	5	4	A,CY ← A + (ES, HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	4	A,CY ← A + ((ES, HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	4	A,CY ← A + ((ES, HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	4	A,CY ← A + ((ES, HL) + C)	×	×	×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash memory area is accessed.

4. Except rp = AX

5. 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (7/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	Note 2	Note 3		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 4]	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	4	A, CY ← A + (ES, addr16) + CY	×	×	×
		A, ES:[HL]	2	2	5	4	A, CY ← A + (ES, HL) + CY	×	×	×
		A, ES:[HL + byte]	3	2	5	4	A, CY ← A + ((ES, HL) + byte) + CY	×	×	×
		A, ES:[HL + B]	3	2	5	4	A, CY ← A + ((ES, HL) + B) + CY	×	×	×
		A, ES:[HL + C]	3	2	5	4	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 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	4	A, CY ← A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	4	A, CY ← A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	4	A, CY ← A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	4	A, CY ← A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	4	A, CY ← A – ((ES:HL) + C)	×	×	×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash memory area is accessed.

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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (8/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	Note 2	Note 3		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 4]	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	4	A, CY ← A – (ES:addr16) – CY	×	×	×
		A, ES:[HL]	2	2	5	4	A, CY ← A – (ES:HL) – CY	×	×	×
		A, ES:[HL + byte]	3	2	5	4	A, CY ← A – ((ES:HL) + byte) – CY	×	×	×
		A, ES:[HL + B]	3	2	5	4	A, CY ← A – ((ES:HL) + B) – CY	×	×	×
		A, ES:[HL + C]	3	2	5	4	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 4]	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	4	A ← A ∧ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	4	A ← A ∧ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	4	A ← A ∧ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	4	A ← A ∧ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	4	A ← A ∧ ((ES:HL) + C)	×		

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash memory area is accessed.

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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (9/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	**Note 2**	**Note 3**		Z	AC	CY
8-bit operation	OR	A, #byte	2	1	–	–	A ← A ∨ byte	×		
		saddr, #byte	3	2	–	–	(saddr) ← (saddr) ∨ byte	×		
		A, r Note 4	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	4	A ← A ∨ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	4	A ← A ∨ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	4	A ← A ∨ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	4	A ← A ∨ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	4	A ← A ∨ ((ES:HL) + C)	×		
	XOR	A, #byte	2	1	–	–	A ← A ⊻ byte	×		
		saddr, #byte	3	2	–	–	(saddr) ← (saddr) ⊻ byte	×		
		A, r Note 4	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	4	A ← A ⊻ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	4	A ← A ⊻ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	4	A ← A ⊻ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	4	A ← A ⊻ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	4	A ← A ⊻ ((ES:HL) + C)	×		

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash memory area is accessed.

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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (10/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	**Note 2**	**Note 3**		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	–	–	A – byte	×	×	×
		saddr, #byte	3	1	–	–	(saddr) – byte	×	×	×
		A, r [Note 4]	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	4	A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	4	A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	4	A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	4	A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	4	A – ((ES:HL) + C)	×	×	×
		ES:!addr16, #byte	5	2	5	4	(ES:addr16) – byte	×	×	×
	CMP0	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	4	–	(addr16) – 00H	×	×	×
		ES:!addr16	4	2	5	4	(ES:addr16) – 00H	×	×	×
	CMPS	X, [HL + byte]	3	1	4	–	X – (HL + byte)	×	×	×
		X, ES:[HL + byte]	4	2	5	4	X – ((ES:HL) + byte)	×	×	×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash memory area is accessed.

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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (11/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	**Note 2**	**Note 3**		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	4	AX, CY ← AX + (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	4	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	4	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	4	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	4	AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	4	AX – ((ES:HL) + byte)	×	×	×
Multiply	MULU	X	1	1	–	–	AX ← A × X			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (12/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	**Note 2**	**Note 3**		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, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

3. cnt indicates the bit shift count.

Table 28-5. Operation List (13/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	Note 2	Note 3		Z	AC	CY
Rotate	ROR	A, 1	2	1	–	–	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1			×
	ROL	A, 1	2	1	–	–	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1			×
	RORC	A, 1	2	1	–	–	(CY ← A_0, A_7 ← CY, $A_{m-1} \leftarrow A_m$) × 1			×
	ROLC	A, 1	2	1	–	–	(CY ← A_7, A_0 ← CY, $A_{m+1} \leftarrow A_m$) × 1			×
	ROLWC	AX,1	2	1	–	–	(CY ← AX_{15}, AX_0 ← CY, $AX_{m+1} \leftarrow AX_m$) × 1			×
		BC,1	2	1	–	–	(CY ← BC_{15}, BC_0 ← 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	4	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	4	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	4	CY ← CY ∨ (ES, HL).bit			×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (14/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	**Note 2**	**Note 3**		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	4	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 ← $\overline{\text{CY}}$			×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (15/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	**Note 2**	**Note 3**		Z	AC	CY
Call/ return	CALL	rp	2	3	–	–	(SP – 2) ← (PC + 2)$_S$, (SP – 3) ← (PC + 2)$_H$, (SP – 4) ← (PC + 2)$_L$, PC ← CS, rp, SP ← SP – 4			
		$!addr20	3	3	–	–	(SP – 2) ← (PC + 3)$_S$, (SP – 3) ← (PC + 3)$_H$, (SP – 4) ← (PC + 3)$_L$, PC ← PC + 3 + jdisp16, SP ← SP – 4			
		!addr16	3	3	–	–	(SP – 2) ← (PC + 3)$_S$, (SP – 3) ← (PC + 3)$_H$, (SP – 4) ← (PC + 3)$_L$, PC ← 0000, addr16, SP ← SP – 4			
		!!addr20	4	3	–	–	(SP – 2) ← (PC + 4)$_S$, (SP – 3) ← (PC + 4)$_H$, (SP – 4) ← (PC + 4)$_L$, PC ← addr20, SP ← SP – 4			
	CALLT	[addr5]	2	5	–	–	(SP – 2) ← (PC + 2)$_S$, (SP – 3) ← (PC + 2)$_H$, (SP – 4) ← (PC + 2)$_L$, PC$_S$ ← 0000, PC$_H$ ← (0000, addr5 + 1), PC$_L$ ← (0000, addr5), SP ← SP – 4			
	BRK	–	2	5	–	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 2)$_S$, (SP – 3) ← (PC + 2)$_H$, (SP – 4) ← (PC + 2)$_L$, PC$_S$ ← 0000, PC$_H$ ← (0007FH), PC$_L$ ← (0007EH), SP ← SP – 4, IE ← 0			
	RET	–	1	6	–	–	PC$_L$ ← (SP), PC$_H$ ← (SP + 1), PC$_S$ ← (SP + 2), SP ← SP + 4			
	RETI	–	2	6	–	–	PC$_L$ ← (SP), PC$_H$ ← (SP + 1), PC$_S$ ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R
	RETB	–	2	6	–	–	PC$_L$ ← (SP), PC$_H$ ← (SP + 1), PC$_S$ ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	**Note 2**	**Note 3**		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	$addr20	2	2/4$^{\text{Note 4}}$	–	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4$^{\text{Note 4}}$	–	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4$^{\text{Note 4}}$	–	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4$^{\text{Note 4}}$	–	–	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4$^{\text{Note 4}}$	–	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=0			
	BNH	$addr20	3	2/4$^{\text{Note 4}}$	–	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=1			
	BT	saddr.bit, $addr20	4	3/5$^{\text{Note 4}}$	–	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5$^{\text{Note 4}}$	–	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5$^{\text{Note 4}}$	–	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5$^{\text{Note 4}}$	–	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5$^{\text{Note 4}}$	6/7	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6$^{\text{Note 4}}$	7/8	6/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash memory area is accessed.

4. 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 28-5. Operation List (17/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks			Operation	Flag		
				Note 1	Note 2	Note 3		Z	AC	CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5[Note 4]	–	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5[Note 4]	–	–	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5[Note 4]	–	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5[Note 4]	–	–	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5[Note 4]	6/7	–	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6[Note 4]	7/8	6/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5[Note 4]	–	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5[Note 4]	–	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5[Note 4]	–	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5[Note 4]	–	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5[Note 4]	–	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6[Note 4]	–	–	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, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. When the data flash memory area is accessed.

4. 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. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

3. n indicates the number of register banks (n = 0 to 3)

CHAPTER 29 ELECTRICAL SPECIFICATIONS

Cautions 1. The 78K0R/Hx3 has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. The pins mounted depend on the product as follows.

(1) Port functions

Port	78K0R/HC3	78K0R/HE3	78K0R/HF3	78K0R/HG3
	48 Pins	64 Pins	80 Pins	100 Pins
Port 0	P00		P00 to P02	P00 to P03
Port 1	P10 to P17			
Port 3	P30 to P32			
Port 4	P40, P41	P40 to P43	P40 to P47	
Port 5	–	P50 to P53	P50 to P57	
Port 6	P60 to P63		P60 to P67	
Port 7	P70 to P73	P70 to P77		
Port 8	P80 to P87			
Port 9	P90 to P92	P90 to P96	P90 to P97	
Port 10	–			P100 to P107
Port 12	P120 to P125		P120 to P126	P120 to P127
Port 13	P130			
Port 14	P140			
Port 15	–			P150 to P157

(The remaining table is on the next page.)

(2) Non-port functions

Port		78K0R/HC3	78K0R/HE3	78K0R/HF3	78K0R/HG3
		48 Pins	64 Pins	80 Pins	100 Pins
		μPD78F1031 to 78F1035	μPD78F1036 to 78F1040	μPD78F1041 to 78F1045	μPD78F1046 to 78F1050
Power supply, ground		V_{DD}, EV_{DD}, V_{SS}, EV_{SS}, AV_{REF}, AV_{SS}			V_{DD}, EV_{DD0}, EV_{DD1}, V_{SS}, EV_{SS0}, EV_{SS1}, AV_{REF}, AV_{SS}
Regulator		REGC			
Reset		$\overline{\text{RESET}}$			
Clock oscillation		X1, X2, EXCLK, EXCLKS			
Writing to flash memory		FLMD0			
Interrupt		INTP0 to INTP7		INTP0 to INTP8	
Key interrupt		KR0 to KR3	KR0 to KR7		
Timer array unit	TAU0	TI00 to TI07, TO00 to TO07			
	TAU1	TI10 to TI17, TO10 to TO17			
	TAU2	–	TI20 to TI23, TO20 to TO23		TI20 to TI27, TO20 to TO27
Serial array unit	CSI00	$\overline{\text{SCK00}}$, SI00, SO00, $\overline{\text{SSI00}}$			
	CSI01	–	$\overline{\text{SCK01}}$, SI01, SO01, $\overline{\text{SSI01}}$		
	CSI10	$\overline{\text{SCK10}}$, SI10, SO10			
	CSI11	–			$\overline{\text{SCK11}}$, SI11, SO11
	IIC11	SCL11, SDA11			
	UART2	–	TxD2, RxD2, INTPR2		
	IIC20	–	SDA20, SCL20		
LIN-UART		LTxD0, LTxD1, LRxD0, LRxD1, INTPLR0, INTPLR1			
CAN controller		CTxD, CRxD			
A/D converter		ANI00 to ANI11, ADTRG	ANI00 to ANI14, ADTRG	ANI00 to ANI15, ADTRG	ANI00 to ANI23, ADTRG
Clock output		PCL			
Reset output		RESOUT			
Low-voltage detector (LVI)		EXLVI, LVIOUT			
On-chip debug function		TOOL0, TOOL1			

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.5 to +6.5	V
	EV_{DD}, EV_{DD0}, EV_{DD1}	EV_{DD} = EV_{DD0} = EV_{DD1}	−0.5 to +6.5	V
	V_{SS}		−0.5 to +0.3	V
	EV_{SS}, EV_{SS0}, EV_{SS1}	EV_{SS} = EV_{SS0} = EV_{SS1}	−0.5 to +0.3	V
	AV_{REF}		−0.5 to V_{DD} +0.3[Note 1]	V
	AV_{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 2]	V
Input voltage	V_{I1}	P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120 to P127, P140, P150 to P157, EXCLK, EXCLKS, $\overline{\text{RESET}}$, FLMD0	−0.3 to EV_{DD} = EV_{DD0} = EV_{DD1} +0.3 and −0.3 to V_{DD} +0.3[Note 1]	V
	V_{I2}	P60 to P63 (N-ch open-drain)	−0.3 to +6.5	V
	V_{I3}	P80 to P87, P90 to P97, P100 to P107	−0.3 to AV_{REF} +0.3 and −0.3 to V_{DD} +0.3[Note 1]	V
Output voltage	V_{O1}	P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P120 to P127, P130, P140, P150 to P157	−0.3 to EV_{DD} = EV_{DD0} = EV_{DD1} +0.3[Note 1]	V
	V_{O2}	P80 to P87, P90 to P97, P100 to P107	−0.3 to AV_{REF} +0.3	V
Analog input voltage	V_{AN}	ANI0 to ANI23	−0.3 to AV_{REF} +0.3[Note 1] and −0.3 to V_{DD} +0.3[Note 1]	V

Notes 1. Must be 6.5 V or lower.

<R> **2.** 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.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120, P125 to P127, P130, P140, P150 to P157	−10	mA
		Total of all pins −80 mA	P01, P02, P40 to P47, P120, P125 to P127, P150 to P153	−25	mA
			P00, P03, P10 to P17, P30 to P32, P50 to P57, P64 to P67, P70 to P77, P130, P140, P154 to P157	−55	mA
	I_{OH2}	Per pin	P80 to P87, P90 to P97, P100 to P107	−0.5	mA
		Total of all pins		−2	mA
Output current, low	I_{OL1}	Per pin	P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P120, P125 to P127, P130, P140, P150 to P157	30	mA
		Total of all pins 200 mA	P01, P02, P40 to P47, P120, P125 to P127, P150 to P153	60	mA
			P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P130, P140, P154 to P157	140	mA
	I_{OL2}	Per pin	P80 to P87, P90 to P97, P100 to P107	1	mA
		Total of all pins		9	mA
Operating ambient temperature	T_A	In normal operation mode		−40 to +85	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			−65 to +150	°C

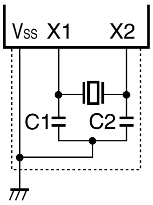
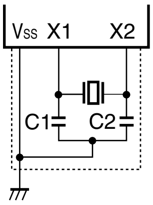
Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

X1 Oscillator Characteristics

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency (f_X)[Note]	AMPH = 0	2.0		10.0	MHz
			AMPH = 1	2.0		20.0	
Crystal resonator		X1 clock oscillation frequency (f_X)[Note]	AMPH = 0	2.0		10.0	MHz
			AMPH = 1	2.0		20.0	

Note Indicates only oscillator characteristics. See 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.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

Internal Oscillator Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Resonator	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Internal high-speed oscillation clock frequency (high-accuracy)	f_{IH8M}	SEL4M = 0	7.84	8.0	8.16	MHz
	f_{IH4M}	SEL4M = 1	3.92	4.0	4.08	MHz
Internal low-speed oscillation clock frequency	f_{IL}		27	30	33	kHz

PLL Circuit Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Resonator	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Frequency of clock that can be input to PLL	f_{PLLI}	When internal high-speed oscillation clock selected	PLLDIV0 = 0	f_{IH4M}			MHz
			PLLDIV0 = 1	f_{IH8M}			MHz
		When high-speed system clock selected	PLLDIV0 = 0	3.92	4.00	4.08	MHz
			PLLDIV0 = 1	7.84	8.00	8.16	MHz
Center value of frequency output from PLL	f_{PLLO}	OPTPLL = 0		$f_{PLLI}/2^{PLLDIV0} \times 8/2^{PLLDIV1}$			MHz
		OPTPLL = 1		$f_{PLLI}/2^{PLLDIV0} \times 6/2^{PLLDIV1}$			MHz
Long-term jitter [Note 1]	T_{LJ}	f_{PLLO} = 24 MHz, 480 count [Note 2]				±2.0	ns
<R>		f_{PLLO} = 16 MHz, 320 count [Note 2]				±2.0	ns

Notes 1. This value applies when the power supply and input clock are stable. It does not include any error that might occur due to fluctuations in the power supply or input clock.

2. This translates to an interval of 20 μs.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

DC Characteristics (1/12)

(T_A = −40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120, P125 to P127, P130, P140, P150 to P157	**Note 3**			−3.0	mA
		Per pin for P00, P03, P10 to P17, P30 to P32, P50 to P57, P64 to P67, P70 to P77, P130, P140, P154 to P157 Among above pins to six of them	**Note 3**			−5.0	mA
		Per pin for P10, P12, P30, P74, P76, P140	**Note 4**			−0.6	mA
		Total of P01, P02, P40 to P47, P120, P125 to P127, P150 to P153 (When duty = 70% or less[Note 2])				−20.0	mA
		Total of P00, P03, P10 to P17, P30 to P32, P50 to P57, P64 to P67, P70 to P77, P130, P140, P154 to P157 (When duty = 70% or less[Note 2])				−44.0	mA
		Total of all pins (When duty = 60% or less[Note 2])				−50.0	mA
	I_{OH2}	Per pin for P80 to P87, P90 to P97, P100 to P107	AV_{REF} = V_{DD}			−0.1	mA
		Total of P80 to P87, P90 to P97, P100 to P107	AV_{REF} = V_{DD}			−2.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from EV_{DD} or EV_{DD0} or EV_{DD1} pin to an output pin. Note, however, that the original duty must be less than n.

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 60% to n%). Note, however, that the original duty must be less than n.

• Total output current of pins = (I_{OH} × 0.6)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = −50.0 mA

Total output current of pins = (−50.0 × 0.6)/(80 × 0.01) = −37.5 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.

3. Slew rate of P10, P12, P30, P74, P76, and P140: Normal mode

4. Slew rate of P10, P12, P30, P74, P76, and P140: Slow mod.

Caution P42, P43, P72, P74, and P76 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.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

DC Characteristics (2/12)

($T_A = -40$ to $+85°C$, $4.0\ V \le V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} \le 5.5\ V$, $AV_{REF} \le V_{DD}$, $V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\ V$)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120, P125 to P127, P130, P140, P150 to P157	**Note 3**			8.5	mA
		Per pin for P60 to P63				15.0	mA
		Per pin for P00, P03, P10 to P17, P30 to P32, P50 to P57, P64 to P67, P70 to P77, P130, P140, P154 to P157 Among above pins to six of them	**Note 3**			10.0	mA
		Per pin for P10, P12, P30, P74, P76, P140	**Note 4**			0.59	mA
		Total of P01, P02, P40 to P47, P120, P125 to P127, P150 to P153 (When duty = 70% or less[Note 2])				24.0	mA
		Total of P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P130, P140, P154 to P157 (When duty = 70% or less[Note 2])				60.0	mA
		Total of all pins (When duty = 60% or less[Note 2])				80.0	mA
	I_{OL2}	Per pin for P80 to P87, P90 to P97, P100 to P107	$AV_{REF} = V_{DD}$			0.4	mA
		Total of P80 to P87, P90 to P97, P100 to P107	$AV_{REF} = V_{DD}$			8.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to EV_{SS}, EV_{SS0}, EV_{SS1}, V_{SS}, and AV_{SS} pin.

2. Specification under conditions where the duty factor is 60% or less or 70% or less.
The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 60% to n%). Note, however, that the original duty must be less than n.

- Total output current of pins = (I_{OL} × 0.6)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 80.0 mA
Total output current of pins = (80.0 × 0.6)/(80 × 0.01) = 60.0 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.

3. Slew rate of P10, P12, P30, P74, P76, and P140: Normal mode

4. Slew rate of P10, P12, P30, P74, P76, and P140: Slow mod.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

DC Characteristics (3/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} < 4.0 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120, P125 to P127, P130, P140, P150 to P157	**Note 3**			–1.0	mA
		Per pin for P10, P12, P30, P74, P76, P140	**Note 4**			–0.2	mA
		Total of P01, P02, P40 to P47, P120, P125 to P127, P150 to P153 (When duty = 70% or less[Note 2])				–10.0	mA
		Total of P00, P03, P10 to P17, P30 to P32, P50 to P57, P64 to P67, P70 to P77, P130, P140, P154 to P157 (When duty = 70% or less[Note 2])				–20.0	mA
		Total of all pins (When duty = 60% or less[Note 2])				–30.0	mA
	I_{OH2}	Per pin for P80 to P87, P90 to P97, P100 to P107	AV_{REF} = V_{DD}			–0.1	mA
		Total of P80 to P87, P90 to P97, P100 to P107	AV_{REF} = V_{DD}			–2.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from EV_{DD} or EV_{DD0} or EV_{DD1} pin to an output pin.

2. Specification under conditions where the duty factor is 60% or less or 70% or less.

The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 60% to n%). Note, however, that the original duty must be less than n.

- Total output current of pins = (I_{OH} × 0.6)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = –30.0 mA

Total output current of pins = (–30.0 × 0.6)/(80 × 0.01) = –22.5 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.

3. Slew rate of P10, P12, P30, P74, P76, and P140: Normal mode

4. Slew rate of P10, P12, P30, P74, P76, and P140: Slow mod.

Caution P42, P43, P72, P74, and P76 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.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

DC Characteristics (4/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} < 4.0 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120, P125 to P127, P130, P140, P150 to P157	**Note 3**			1.0	mA
		Per pin for P10, P12, P30, P74, P76, P140	**Note 4**			0.07	mA
		Per pin for P60 to P63				3.0	mA
		Total of P01, P02, P40 to P47, P120, P125 to P127, P150 to P153 (When duty = 70% or less[Note 2])				21.0	mA
		Total of P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P130, P140, P154 to P157 (When duty = 70% or less[Note 2])				53.0	mA
		Total of all pins (When duty = 60% or less[Note 2])				70.0	mA
	I_{OL2}	Per pin for P80 to P87, P90 to P97, P100 to P107	AV_{REF} = V_{DD}			0.4	mA
		Total of P80 to P87, P90 to P97, P100 to P107	AV_{REF} = V_{DD}			8.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to EV_{SS}, EV_{SS0}, EV_{SS1}, V_{SS}, and AV_{SS} pin.

2. Specification under conditions where the duty factor is 60 % or less or 70% or less.
The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 60% to n%). Note, however, that the original duty must be less than n.

- Total output current of pins = (I_{OL} × 0.6)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 70.0 mA
Total output current of pins = (70.0 × 0.6)/(80 × 0.01) = 52.5 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.

3. Slew rate of P10, P12, P30, P74, P76, and P140: Normal mode

4. Slew rate of P10, P12, P30, P74, P76, and P140: Slow mod.

Caution P42, P43, P72, P74, and P76 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.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

<R> DC Characteristics (5/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P64 to P67, P70 to P72, P74, P120, P125 to P127, P140, P150 to P157	4.0 V ≤ V_{DD} ≤ 5.5 V	$0.65V_{DD}$		V_{DD}	V
			2.7 V ≤ V_{DD} < 4.0 V	$0.7V_{DD}$		V_{DD}	V
	V_{IH2}	P73, P75 to P77	Normal input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	$0.65V_{DD}$		V_{DD}	V
			Normal input buffer 2.7 V ≤ V_{DD} < 4.0 V	$0.7V_{DD}$		V_{DD}	V
	V_{IH3}	P73, P75 to P77	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	2.2		V_{DD}	V
			TTL input buffer 2.7 V ≤ V_{DD} < 4.0 V	2.0		V_{DD}	V
	V_{IH4}	P121 to P124, $\overline{\text{RESET}}$		$0.8V_{DD}$		V_{DD}	V
	V_{IH5}	P80 to P87, P90 to P97, P100 to P107	2.7 V ≤ AV_{REF} ≤ V_{DD} ≤ 5.5 V	$0.8AV_{REF}$		AV_{REF}	V
	V_{IH6}	P60 to P63	Normal input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	$0.65V_{DD}$		6.0	V
			Normal input buffer 2.7 V ≤ V_{DD} < 4.0 V	$0.7V_{DD}$		6.0	V
	V_{IH7}	P60, P61, P63	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	2.2		6.0	V
			TTL input buffer 2.7 V ≤ V_{DD} < 4.0 V	2.0		6.0	V
	V_{IH8}	FLMD0		$0.9V_{DD}$ Note		V_{DD}	V

Note Must be $0.9V_{DD}$ or higher when used in the flash memory programming mode.

Caution The maximum value of V_{IH} of pins P42, P43, P72, P74, and P76 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.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

<R> DC Characteristics (6/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, low	V_{IL1}	P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P64 to P67, P70 to P72, P74, P120, P125 to P127, P140, P150 to P157	4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.35V_{DD}	V
			2.7 V ≤ V_{DD} < 4.0 V	0		0.3V_{DD}	V
	V_{IL2}	P73, P75 to P77	Normal input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.35V_{DD}	V
			Normal input buffer 2.7 V ≤ V_{DD} < 4.0 V	0		0.3V_{DD}	V
	V_{IL3}	P73, P75 to P77	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 2.7 V ≤ V_{DD} < 4.0 V	0		0.5	V
	V_{IL4}	P121 to P124, $\overline{\text{RESET}}$		0		0.2V_{DD}	V
	V_{IL5}	P80 to P87, P90 to P97, P100 to P107	4.0 V ≤ AV_{REF} ≤ V_{DD} ≤ 5.5 V	0		0.5AV_{REF}	V
			2.7 V ≤ AV_{REF} = V_{DD} < 4.0 V	0		0.4AV_{REF}	V
	V_{IL6}	P60 to P63	Normal input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.35V_{DD}	V
			Normal input buffer 2.7 V ≤ V_{DD} < 4.0 V	0		0.3V_{DD}	V
	V_{IL7}	P60, P61, P63	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 2.7 V ≤ V_{DD} < 4.0 V	0		0.5	V
	V_{IL8}	FLMD0 [Note]		0		0.1V_{DD}	V

Note When disabling writing of the flash memory, connect the FLMD0 pin processing directly to V_{SS}, and maintain a voltage less than 0.1V_{DD}.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

DC Characteristics (7/12)
(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120, P125 to P127, P130, P140, P150 to P157	**Note 1**	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = − 3.0 mA	V_{DD} − 0.7			V
				2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = −1.0 mA	V_{DD} − 0.5			V
	V_{OH2}	P80 to P87, P90 to P97, P100 to P107		AV_{REF} = V_{DD}, I_{OH2} = −0.1 mA	AV_{REF} − 0.5			V
	V_{OH3}	P00, P03, P10 to P17, P30 to P32, P50 to P57, P64 to P67, P70 to P77, P130, P140, P154 to P157 Among above pins to six of them	**Note 1**	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = − 5.0 mA	V_{DD} − 1.0			V
	V_{OH4}	P10, P12, P30, P74, P76, P140	**Note 2**	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = − 0.6 mA	V_{DD} − 0.8			V
				2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = −0.2 mA	V_{DD} − 0.5			V
Output voltage, low	V_{OL1}	P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120 to P127, P130, P140, P150 to P157	**Note 1**	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
				2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.0 mA			0.5	V
	V_{OL2}	P80 to P87, P90 to P97, P100 to P107		AV_{REF} = V_{DD}, I_{OL2} = 0.4 mA			0.4	V
	V_{OL3}	P60 to P63		4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 15.0 mA			2.0	V
				4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 5.0 mA			0.4	V
				2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.4	V
	V_{OL4}	P00, P03, P10 to P17, P30 to P32, P50 to P57, P64 to P67, P70 to P77, P130, P140, P154 to P157 Among above pins to six of them	**Note 1**	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 10.0 mA			1.0	V
	V_{OL5}	P10, P12, P30, P74, P76, P140	**Note 2**	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.59 mA			0.8	V
				2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.07 mA			0.5	V

Notes 1. Slew rate of P10, P12, P30, P74, P76, and P140: Normal mode

2. Slew rate of P10, P12, P30, P74, P76, and P140: Slow mod.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

<R> **DC Characteristics (8/12)**

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120 to P127, P140, P150 to P157, FLMD0, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P80 to P87, P90 to P97, P100 to P107	$V_I = AV_{REF}$, $AV_{REF} = V_{DD}$				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, EXCLK, EXCLKS)	$V_I = V_{DD}$	In Input port			1	μA
				In external clock mode (EXCLK, EXCLKS)			1	μA
				In resonator connection (X1, X2)			10	μA
	I_{LIH4}	P60 to P63	$V_{DD} < V_I$ ≤ 6.0 V	On-chip pull-up resistor connected			5	μA
				On-chip pull-up resistor not connected			1	μA
Input leakage current, low	I_{LIL1}	P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P120 to P127, P140, P150 to P157, FLMD0, $\overline{\text{RESET}}$	$V_I = V_{SS}$				–1	μA
	I_{LIL2}	P80 to P87, P90 to P97, P100 to P107	$V_I = V_{SS}$, $AV_{REF} = V_{DD}$				–1	μA
	I_{LIL3}	P121 to P124 (X1, X2, EXCLK, EXCLKS)	$V_I = V_{SS}$	In Input port			–1	μA
				In external clock mode (EXCLK, EXCLKS)			–1	μA
				In resonator connection (X1, X2)			–10	μA

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

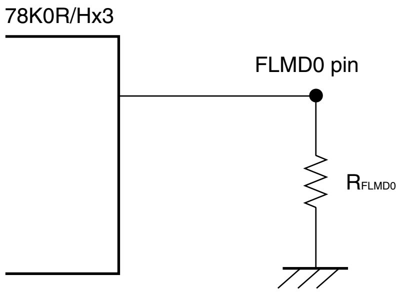
Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

DC Characteristics (9/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
On-chip pll-up resistance	R_{U1}	P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P120, P125 to P127, P140, P150 to P157	V_I = V_{SS}, In input port	10	20	40	kΩ
	R_{U2}	P60 to P63	V_I = V_{SS}	15	27	50	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.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

<R>

DC Characteristics (10/12)

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current	I_{DD1}[Note 1]	Operating mode	f_{CLK} = 24 MHz[Notes 2, 3, 5]	f_{MX} = 8 MHz, f_{CLK} = f_{MX} × 3	Square wave input		9.5	20	mA
					Resonator connection		9.6	20	
				f_{MX} = 8 MHz, f_{CLK} = f_{IH} × 3, aFCAN stopped			9.5	18	
				f_{MX} = 4 MHz, f_{CLK} = f_{IH} × 6, aFCAN stopped			9.5	18	
			f_{CLK} = 20 MHz[Notes 2, 3, 5]	f_{MX} = 20 MHz, f_{CLK} = f_{MX}, PLL stopped, aFCAN stopped	Square wave input		7.5	15	mA
					Resonator connection		7.8	15	
			f_{CLK} = 10 MHz[Notes 2, 3, 5]	f_{MX} = 10 MHz, f_{CLK} = f_{MX}, PLL stopped, aFCAN stopped	Square wave input		4.1	8.0	mA
					Resonator connection		4.2	8.0	
			f_{CLK} = 8 MHz[Notes 2, 3, 5]	f_{MX} = 8 MHz, f_{CLK} = f_{MX}, PLL stopped	Square wave input		3.2	7.5	mA
					Resonator connection		3.4	7.5	
				f_{IH} = 8 MHz, f_{CLK} = f_{IH}, PLL stopped, aFCAN stopped			3.3	6.5	
			f_{MX} = 4 MHz[Notes 2, 3, 5]	f_{MX} = 4 MHz, f_{CLK} = f_{MX}, PLL stopped, aFCAN stopped	Square wave input		1.7	3.5	mA
					Resonator connection		1.8	3.5	
				f_{IH} = 4 MHz, f_{CLK} = f_{IH}, PLL stopped, aFCAN stopped			1.7	3.5	
			f_{CLK} = f_{IL}	PLL stopped, aFCAN stopped	T_A = −40 to +70°C[Notes 2, 3, 5]		12	40	μA
					T_A = −40 to +85°C[Notes 3, 4, 5]		14	50	

Notes 1. Total current flowing into V_{DD}, EV_{DD}, EV_{DD0}, EV_{DD1}, and AV_{REF}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.

2. The typical value is that when T_A = +25°C and V_{DD} = 5.0 V. Peripheral devices and the data flash are stopped.

3. The maximum value is that when all peripheral devices are operating. The A/D converter, LVI circuit, and data flash are stopped and the code flash is being read. The 16-bit wakeup timer operates with f_{IL}.

4. The typical value is that when T_A = +70°C and V_{DD} = 5.0 V. Peripheral devices and the data flash are stopped.

5. Clocks other than those selected as f_{IL} and f_{CLK} are stopped and the code flash is being read. The current flowing to the I/O buffers is not included.

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

2. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

3. f_{IH}: Internal high-speed oscillation clock frequency

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

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

<R>

DC Characteristics (11/12)

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current	I_{DD2}[Note 1]	HALT mode	f_{CLK} = 24 MHz[Notes 2, 3,]	f_{MX} = 8 MHz, f_{CLK} = f_{MX} × 3	Square wave input		2.6	12.0	mA
					Resonator connection		2.8	12.0	
				f_{MX} = 8 MHz, f_{CLK} = f_{IH} × 3, aFCAN stopped			2.7	10.0	
				f_{MX} = 4 MHz, f_{CLK} = f_{IH} × 6, aFCAN stopped			2.7	10.0	
			f_{CLK} = 20 MHz[Notes 2, 3]	f_{MX} =20 MHz, f_{CLK} = f_{MX}, PLL stopped, aFCAN stopped	Square wave input		1.6	8.0	mA
					Resonator connection		2.0	8.0	
			f_{CLK} = 10 MHz[Notes 2, 3]	f_{MX} = 10 MHz, f_{CLK} = f_{MX}, PLL stopped, aFCAN stopped	Square wave input		0.8	4.0	mA
					Resonator connection		1.0	4.0	
			f_{CLK} = 8 MHz[Notes 2, 3]	f_{MX} = 8 MHz, f_{CLK} = f_{MX}, PLL stopped	Square wave input		0.7	4.5	mA
					Resonator connection		0.8	4.5	
				f_{MX} = 8 MHz, f_{CLK} = f_{IH}, PLL stopped, aFCAN stopped			0.7	3.5	
			f_{MX} = 4 MHz[Notes 2, 3]	f_{MX} = 4 MHz, f_{CLK} = f_{MX}, PLL stopped, aFCAN stopped	Square wave input		0.4	2.0	mA
					Resonator connection		0.5	2.0	
				f_{MX} = 4 MHz, f_{CLK} = f_{IH}, PLL stopped, aFCAN stopped			0.45	2.0	
			f_{CLK} = f_{IL}	PLL stopped, aFCAN stopped	T_A = −40 to +70°C[Notes 2, 3,]		3.5	25	μA
					T_A = −40 to +85°C[Notes 3, 4]		4.7	35	
	I_{DD3}[Note 5]	STOP mode	T_A = −40 to +70 °C[Notes 6, 7]				1.0	15	μA
			T_A = −40 to +85 °C[Notes 7, 8]				2.5	25	μA

Notes

1. Total current flowing into V_{DD}, EV_{DD}, EV_{DD0}, EV_{DD1}, and AV_{REF} including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. During HALT instruction execution by flash memory.

2. The typical value is that when T_A = +25°C and V_{DD} = 5.0 V. Peripheral devices, the data flash, and clocks other than those selected as f_{IL} and f_{CLK} are stopped. The current flowing to the I/O buffers is not included.

3. The maximum value is that when all peripheral devices are operating. The A/D converter, LVI circuit, data flash, and clocks other than those selected as f_{IL} and f_{CLK} are stopped and the code flash is being read. The current flowing to the I/O buffers is not included. The 16-bit wakeup timer operates with f_{IL}.

4. The typical value is that when T_A = +70°C and V_{DD} = 5.0 V. Peripheral devices, the data flash, and clocks other than those selected as f_{IL} and f_{CLK} are stopped. The current flowing to the I/O buffers is not included.

5. This is the total current flowing to V_{DD}, EV_{DD}, EV_{DD0}, EV_{DD1}, and AV_{REF}. It includes the input leakage current when the input pin is fixed to V_{DD} or V_{SS}. The subclock and watchdog timer are stopped.

(**Notes** and **Remarks** are given on the next page.)

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

6. The typical value is that when T_A = +25°C and V_{DD} = 5.0 V. Peripheral devices and f_{IL} are stopped. The current flowing to the I/O buffers is not included.

7. The maximum value is that when all peripheral devices are operating. However, clocks other than an A/D converter, a LVI circuit, and f_{IL} stop, and a code flash is in a reading state. The current flowing to the I/O buffers is not included. The 16-bit wakeup timer operates with f_{IL}.

8. The typical value is that when T_A = +70°C and V_{DD} = 5.0 V. Peripheral devices and f_{IL} are stopped. The current flowing to the I/O buffers is not included.

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

2. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

3. f_{IH}: Internal high-speed oscillation clock frequency

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

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

DC Characteristics (12/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Watchdog timer operating current	I_{WDT} [Notes 1, 2, 3]	f_{IL} = 30 kHz, During STOP mode				0.52	1.1	μA
A/D converter operating current	I_{ADC}[Note 4]	During conversion at maximum speed	High-speed mode 1	AV_{REF} = V_{DD} = 5.0 V		1.72	3.2	mA
			High-speed mode 2	AV_{REF} = V_{DD} = 3.0 V		0.72	1.6	
			Normal mode	AV_{REF} = V_{DD} = 5.0 V		0.86	1.9	
LVI operating current	I_{LVI}[Note 5]					9	18	μA
Data flash operating current	I_{DFL}[Notes 1, 6]	DFLEN = 0					0	mA
		DFLEN = 1	When STOP mode				0	
			When HALT mode			0.2	0.4	
			When waiting (read address is outside data flash)			0.3	0.5	
			When reading (read address is within data flash)				11.0[Note 7]	

Notes 1. The current flows to the V_{DD} pin.

2. When internal high-speed oscillator and high-speed system clock are stopped.

3. Current flowing only to the watchdog timer (including the operating current of the internal low-speed oscillator). The current value of the 78K0R/Hx3 is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when f_{CLK} = f_{IL}/2 or when the watchdog timer operates in STOP mode.

4. Current flowing only to the A/D converter (AV_{REF} pin). The current value of the 78K0R/Hx3 is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

5. Current flowing only to the LVI circuit. The current value of the 78K0R/Hx3 is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVI circuit operates in the Operating, HALT or STOP mode

6. The leakage current is not included.

7. A read access to the data flash memory require minimum four clock cycles, however due to the short peak for this maximum current the average can be calculated by 1/4 of the specified value.

Remarks 1. f_{IL}: Internal low-speed oscillation clock frequency

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

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

AC Characteristics

(1) Basic operation (1/4)

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{XP}) operation	FSEL =0		0.1		8.16	μs
			FSEL =1	When PLL is not used	0.05		8.16	μs
<R>				When PLL is used	0.04		2.1	μs
		Internal low-speed oscillator operation			30	33.3	37	μs
		In the self programming mode			**Note 1**		0.5	μs
External main system clock frequency	f_{EX}				2.0		20.0	MHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}				24.0			ns
External subclock input frequency	f_{EXS}				30.0		1000	kHz
External subclock input high-level width, low-level width	t_{EXSH}, t_{EXSL}				480			ns
TImn input high-level width, low-level width	t_{TIH}, t_{TIL}				$1/f_{MCK}$+ 10			ns
TOmn output frequency	f_{TO}	C = 30 pF	TO00/P10, TO01/P30, TO16/P12	Normal mode (PSRSEL.PSRk = 0)			12	MHz
				Slow mode (PSRSEL.PSRk = 1)			2	
			TOmn other than above	4.0 V ≤ V_{DD} ≤ 5.5 V			12 Note2	
				2.7 V ≤ V_{DD} < 4.0 V			6 Note2	
PCLB output frequency	f_{PCL}	C = 30 pF	Normal mode (PSRSEL.PSRk = 0)				12	MHz
			Slow mode (PSRSEL.PSRk = 1)				2	
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}				1			μs
Key interrupt input low-level width	t_{KR}				250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

Notes 1. The minimum value complies with the conditions when operating on the main system clock. The self-programming operation when the internal low-speed oscillator is operating is not guaranteed.

2. When in normal mode (PSRSEL.PSRk = 0)

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 to 2), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 17, 20 to 27)

2. k = 10, 12, 30, 74, 76, 140

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(1) Basic operation (2/4)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

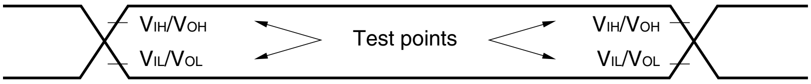
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output rise time, fall time	t_{RO}, t_{FO}	P00 to P03, P11, P13 to P17, P31, P32, P42 to P47, P50 to P57, P64 to P67, P70 to P73, P75, P77, P120, P125 to P127, P130, P150 to P157	Slew rate: Normal mode, C = 30 pF	4.0 V ≤ V_{DD} ≤ 5.5 V			25	ns
				2.7 V ≤ V_{DD} < 4.0 V			55	ns
		P10, P12, P30, P40, P41, P74, P76, P140	Slew rate: Normal mode, C = 30 pF				25	ns
		P10, P12, P30, P74, P76, P140	Slew rate: Slow mode, C = 30 pF	4.0 V ≤ V_{DD} ≤ 5.5 V		25[Note]	60	ns
				2.7 V ≤ V_{DD} < 4.0 V			100	ns

Note When T_A = +25°C and V_{DD} = 5.0 V

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

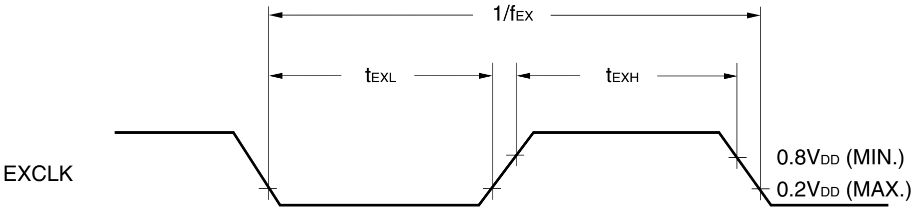
(1) Basic operation (3/4)

AC Timing Test Points[Note]

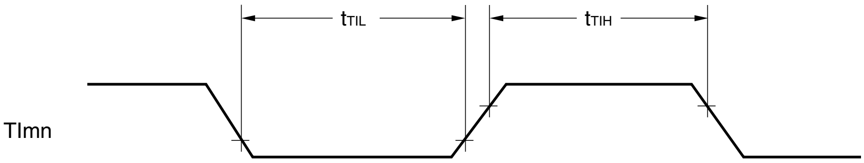


Note t_{RO} and t_{FO} (the output rise and fall times) are measured when $V_{OL} = 0.1V_{DD}$ and $V_{OH} = 0.9V_{DD}$. For the V_{OL}, V_{OH}, V_{IL}, and V_{IH} values at which parameters other than t_{RO} and t_{FO} are measured, see the DC characteristics.

External Main System Clock Timing

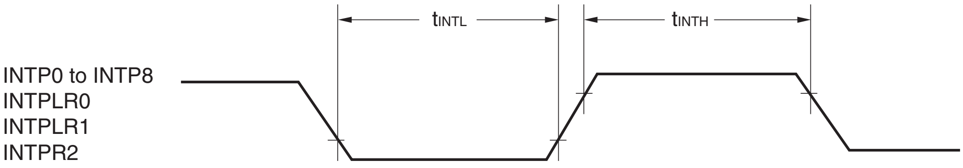


TI Timing

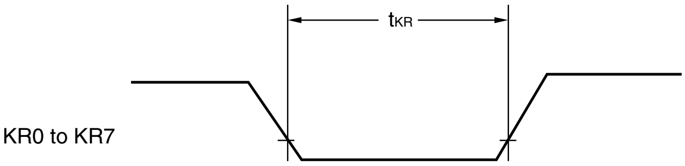


Remark n: Channel number, m: Unit number, mn = 00 to 07, 10 to 17, 20 to 27

<R> **Interrupt Request Input Timing**



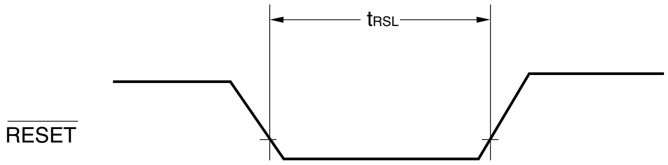
Key Interrupt Input Timing



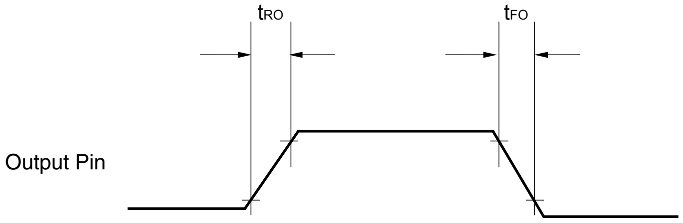
Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(1) Basic operation (4/4)

$\overline{\text{RESET}}$ Input Timing



Output Rise Time, Fall Time Timing



Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

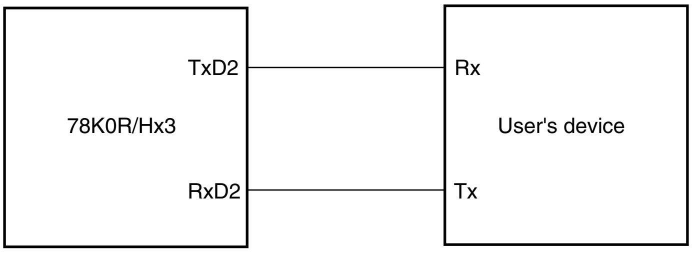
(2) Serial interface: Serial array unit (1/27)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

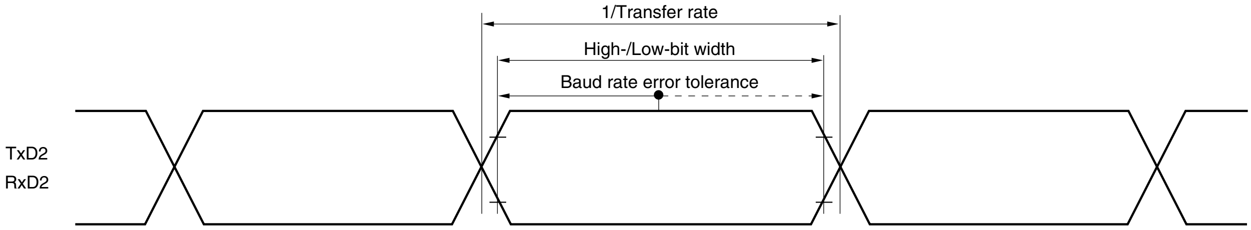
(a) During communication at same potential (UART mode) (dedicated baud rate generator output)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Transfer rate						f_{MCK}/6	bps
		f_{CLK} = 24 MHz, f_{MCK} = f_{CLK},	Slew rate: Normal mode			4	Mbps
			Slew rate: Slow mode			2	Mbps

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Caution Select the normal output mode for TxD2 by using the POM4 register.

Remark f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS2n bit of the SMR2n register. n: Channel number (n = 0, 1))

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (2/27)

(b) During communication at same potential (CSI mode) (master mode, $\overline{\text{SCKp}}$... internal clock output, slew rate: normal mode)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time		4.0 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V	200[Note 5]			ns
		2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} < 4.0 V	300[Note 5]			ns
$\overline{\text{SCKp}}$ high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V	t_{KCY1}/2 – 20			ns
		2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} < 4.0 V	t_{KCY1}/2 – 35			ns
SIp setup time (to $\overline{\text{SCKp}}$↑)[Note 1]	t_{SIK1}	4.0 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V	70			ns
		2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} < 4.0 V	100			ns
SIp hold time (from $\overline{\text{SCKp}}$↑)[Note 2]	t_{KSI1}		30			ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp output[Note 3]	t_{KSO1}	C = 30 pF[Note 4]			40	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{\text{SCKp}}$↓" 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 $\overline{\text{SCKp}}$↓" 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 $\overline{\text{SCKp}}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the $\overline{\text{SCKp}}$ and SOp output lines.

5. And t_{KCY1} ≥ 4/f_{CLK}.

Cautions 1. For CSI00, this is the value when P15 to P17 is selected. (For details, see 11.3 (15) Serial communication pin select register (STSEL).)

2. Select the normal input buffer for SIj and the normal output mode for SOj and $\overline{\text{SCKj}}$ by using the PIMg and POM7 registers.

Remarks 1. p: CSI number (p = 00, 01, 10, 11), g: PIM and POM number (g = 6, 7),
j: CSI number for which communication at different potential can be selected (j = 00, 01)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1))

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (3/27)

(c) During communication at same potential (CSI mode) (master mode, $\overline{\text{SCKp}}$... internal clock output, slew rate: slow mode)

(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t_{KCY1}		500[Note 5]			ns
$\overline{\text{SCKp}}$ high-/low-level width	t_{KH1}, t_{KL1}		$t_{KCY1}/2 - 60$			ns
SIp setup time (to $\overline{\text{SCKp}}$↑)[Note 1]	t_{SIK1}		120			ns
SIp hold time (from $\overline{\text{SCKp}}$↑)[Note 2]	t_{KSI1}		80			ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp output[Note 3]	t_{KSO1}	C = 30 pF[Note 4]			90	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{\text{SCKp}}$↓" 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 $\overline{\text{SCKp}}$↓" 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 $\overline{\text{SCKp}}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the $\overline{\text{SCKp}}$ and SOp output lines.

5. And $t_{KCY1} \geq 4/f_{CLK}$.

Cautions 1. For CSI00, this is the value when P15 to P17 is selected. (For details, see 11.3 (15) Serial communication pin select register (STSEL).)

2. Select the normal input buffer for SIj and the normal output mode for SOj and $\overline{\text{SCKj}}$ by using the PIMg and POM7 registers.

Remarks 1. p: CSI number (p = 00, 01, 10, 11), g: PIM and POM number (g = 6, 7),
j: CSI number for which communication at different potential can be selected (j = 00, 01)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (4/27)

(d) During communication at same potential (CSI mode) (slave mode, $\overline{\text{SCKp}}$... external clock input, slew rate: normal mode)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t_{KCY2}	4.0 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V	20 MHz < f_{MCK}	8/f_{MCK}			ns
			f_{MCK} ≤ 20 MHz	6/f_{MCK}			ns
		2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} < 4.0 V	16 MHz < f_{MCK}	8/f_{MCK}			ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}			ns
$\overline{\text{SCKp}}$ high-/low-level width	t_{KH2}, t_{KL2}			t_{KCY2}/2			ns
SIp setup time (to $\overline{\text{SCKp}}$↑)[Note 1]	t_{SIK2}			80			ns
SIp hold time (from $\overline{\text{SCKp}}$↑)[Note 2]	t_{KSI2}			1/f_{MCK}+50			ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp output[Note 3]	t_{KSO2}	C = 30 pF[Note 4]	4.0 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V			2/f_{MCK}+45	ns
			2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} < 4.0 V			2/f_{MCK}+57	ns
$\overline{\text{SSIp}}$ setup time	t_{SSIK}	DAP = 0		120			ns
		DAP = 1		1/f_{MCK}+120			ns
$\overline{\text{SSIp}}$ hold time	t_{KSSL}	DAP = 0		1/f_{MCK}+120			ns
		DAP = 1		120			ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{\text{SCKp}}$↓" 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 $\overline{\text{SCKp}}$↓" 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 $\overline{\text{SCKp}}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SOp output lines.

Cautions 1. For CSI00, this is the value when P15 to P17 and P30 is selected. (For details, see 11.3 (15) Serial communication pin select register (STSEL).)

2. Select the normal input buffer for SIj and $\overline{\text{SCKj}}$ and the normal output mode for SOj by using the PIMg and POM7 registers.

Remarks 1. p: CSI number (p = 00, 01, 10, 11), g: PIM and POM number (g = 6, 7),
j: CSI number for which communication at different potential can be selected (j = 00, 01)

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, 1))

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (5/27)

(e) During communication at same potential (CSI mode) (slave mode, $\overline{\text{SCKp}}$... external clock input, slew rate: slow mode)

(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EVDD = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t_{KCY2}	20 MHz < f_{MCK}	10/f_{MCK}			ns
		10 MHz < f_{MCK} ≤ 20 MHz	8/f_{MCK}			ns
		f_{MCK} ≤ 10 MHz	6/f_{MCK}			ns
$\overline{\text{SCKp}}$ high-/low-level width	t_{KH2}, t_{KL2}		t_{KCY2}/2			ns
SIp setup time (to $\overline{\text{SCKp}}$↑)[Note 1]	t_{SIK2}		80			ns
SIp hold time (from $\overline{\text{SCKp}}$↑)[Note 2]	t_{KSI2}		1/f_{MCK}+50			ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp output[Note 3]	t_{KSO2}	C = 30 pF[Note 4]			2/f_{MCK}+80	ns
$\overline{\text{SSIp}}$ setup time	t_{SSIK}	DAP = 0	120			ns
		DAP = 1	1/f_{MCK}+120			ns
$\overline{\text{SSIp}}$ hold time	t_{KSSL}	DAP = 0	1/f_{MCK}+120			ns
		DAP = 1	120			ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to $\overline{\text{SCKp}}$↓” 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 $\overline{\text{SCKp}}$↓” 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 $\overline{\text{SCKp}}$↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SOp output lines.

Cautions 1. For CSI00, this is the value when P15 to P17 and P30 is selected. (For details, see 11.3 (15) Serial communication pin select register (STSEL).)

2. Select the normal input buffer for SIj and $\overline{\text{SCKj}}$ and the normal output mode for SOj by using the PIMg and POM7 registers.

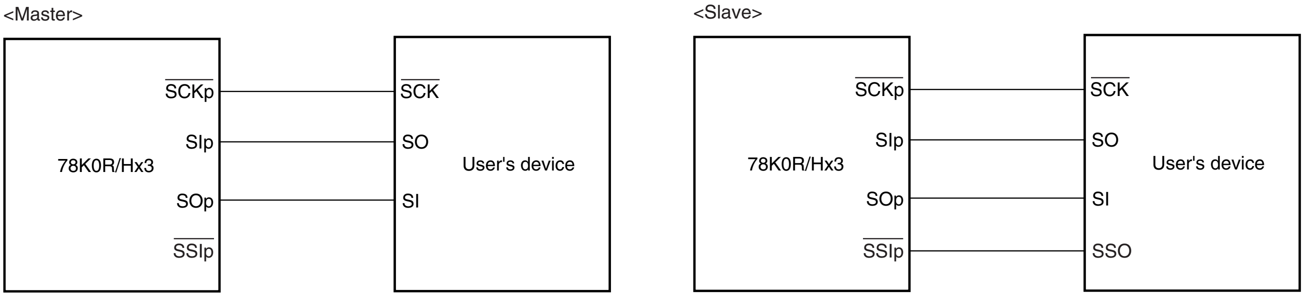
Remarks 1. p: CSI number (p = 00, 01, 10, 11), g: PIM and POM number (g = 6, 7),
j: CSI number for which communication at different potential can be selected (j = 00, 01)

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, 1))

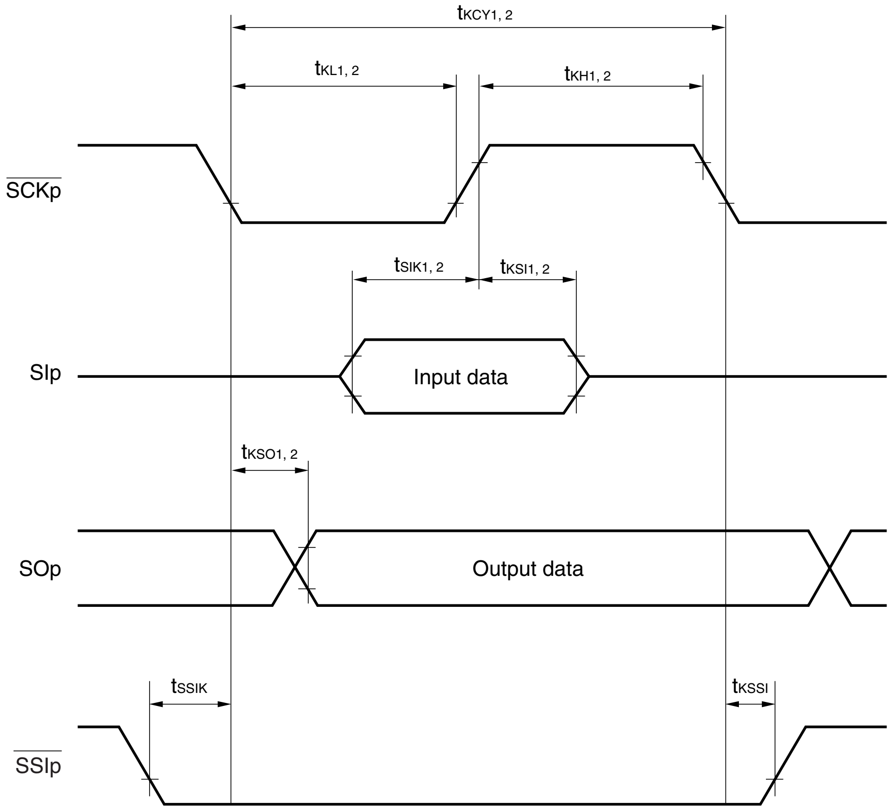
Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (6/27)

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.)



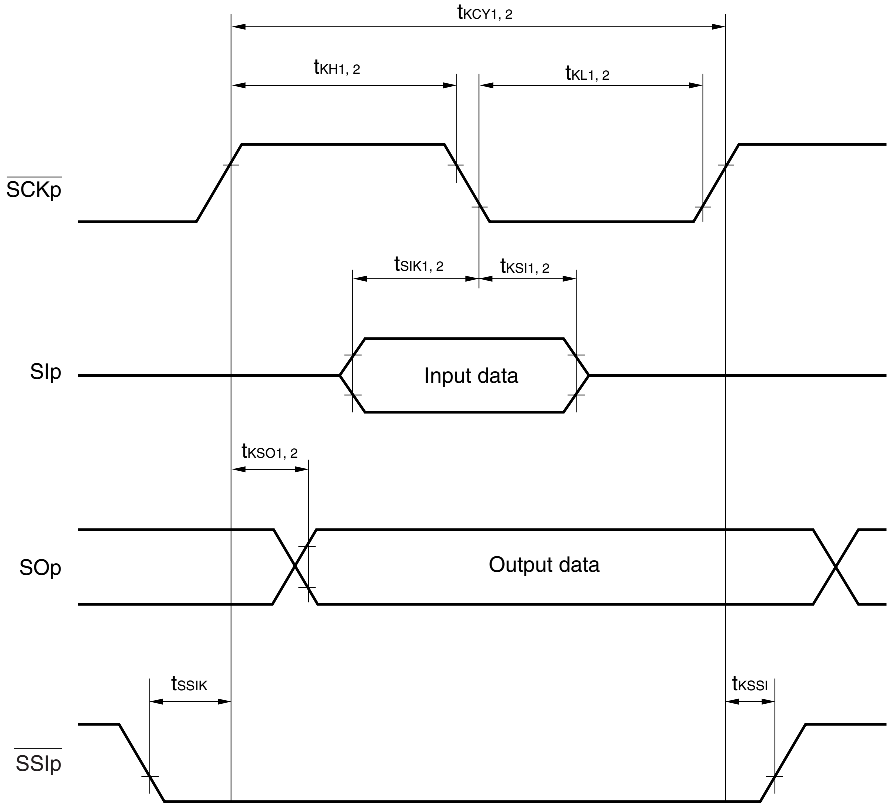
Remarks 1. p: CSI number (p = 00, 01, 10, 11)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (7/27)

CSI mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remarks 1. p: CSI number (p = 00, 01, 10, 11)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(2) Serial interface: Serial array unit (8/27)

(f) During communication at same potential by CSI00 (when using P60 to P63) (CSI mode) (master mode, $\overline{\text{SCKp}}$... internal clock output, slew rate: normal mode)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

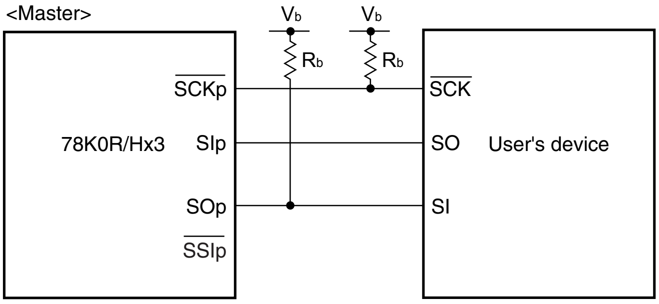
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.1 kΩ	400[Note 3]		ns
		V_{DD} < 4.0 V, V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.8 kΩ			
$\overline{\text{SCKp}}$ high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.1 kΩ	t_{KCY1}/2–40		ns
		V_{DD} < 4.0 V, V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.8 kΩ	t_{KCY1}/2–70		ns
$\overline{\text{SCKp}}$ low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.1 kΩ	t_{KCY1}/2–20		ns
		V_{DD} < 4.0 V, V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.8 kΩ	t_{KCY1}/2–35		ns
SIp setup time (to $\overline{\text{SCKp}}$↑)[Note 1]	t_{SIK1}	4.0 V ≤ V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.1 kΩ	115		ns
		V_{DD} < 4.0 V, V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.8 kΩ	145		ns
SIp setup time (to $\overline{\text{SCKp}}$↓)[Note 2]	t_{SIK1}	4.0 V ≤ V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.1 kΩ	70		ns
		V_{DD} < 4.0 V, V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.8 kΩ	70		ns
SIp hold time (from $\overline{\text{SCKp}}$↑)[Note 1]	t_{KSI1}	4.0 V ≤ V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.1 kΩ	30		ns
		V_{DD} < 4.0 V, V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.8 kΩ	30		ns
SIp hold time (from $\overline{\text{SCKp}}$↓)[Note 2]	t_{KSI1}	4.0 V ≤ V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.1 kΩ	30		ns
		V_{DD} < 4.0 V, V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.8 kΩ	30		ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp output[Note 1]	t_{KSO1}	4.0 V ≤ V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.1 kΩ		85	ns
		V_{DD} < 4.0 V, V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.8 kΩ		115	ns
Delay time from $\overline{\text{SCKp}}$↑ to SOp output[Note 2]	t_{KSO1}	4.0 V ≤ V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.1 kΩ		40	ns
		V_{DD} < 4.0 V, V_{DD} ≤ V_b, C_b = 30 pF, R_b = 1.8 kΩ		40	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. And t_{KCY1} ≥ 4/t_{CLK}.

CSI mode connection diagram (communication at same potential by CSI00 (when using P60 to P63))



Cautions 1. Select the normal input buffer mode for SIp by using the PIM6 register. SOp and $\overline{\text{SCKp}}$ are fixed to the N-ch open drain output (6 V tolerance) mode.

2. The timing diagram showing data transfer between pins with the same potential via CSI00 (when using P60 to P63) is the same as the diagram showing CSI transfer between pins with a different potential.

Remarks 1. p: CSI number (p = 00)

2. m: Unit number (m = 0), n: Channel number (n = 0)

3. R_b[Ω]: Communication line ($\overline{\text{SCKp}}$, SOp) pull-up resistance,
C_b[F]: Communication line (SOp, $\overline{\text{SCKp}}$) load capacitance, V_b[V]: Communication line voltage

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (9/27)

(g) During communication at same potential by CSI00 (when using P60 to P63) (CSI mode) (slave mode, $\overline{\text{SCKp}}$... external clock input, slew rate: normal mode)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t_{KCY2}	4.0 V ≤ V_{DD} ≤ V_b	20 MHz < f_{MCK}	10/f_{MCK}		ns
			10 MHz < f_{MCK} ≤ 20 MHz	8/f_{MCK}		ns
			f_{MCK} ≤ 10 MHz	6/f_{MCK}		ns
		V_{DD} < 4.0 V, V_{DD} ≤ V_b	23.2 MHz < f_{MCK}	22/f_{MCK}		ns
			20.3 MHz < f_{MCK} ≤ 23.2 MHz	20/f_{MCK}		ns
			17.4 MHz < f_{MCK} ≤ 20.3 MHz	18/f_{MCK}		ns
			14.5 MHz < f_{MCK} ≤ 17.4 MHz	16/f_{MCK}		ns
			11.6 MHz < f_{MCK} ≤ 14.5 MHz	14/f_{MCK}		ns
			8.7 MHz < f_{MCK} ≤ 11.6 MHz	12/f_{MCK}		ns
			5.8 MHz < f_{MCK} ≤ 8.7 MHz	10/f_{MCK}		ns
			2.9 MHz <f_{MCK} ≤ 5.8 MHz	8/f_{MCK}		ns
			f_{MCK} ≤ 2.9 MHz	6/f_{MCK}		ns
$\overline{\text{SCKp}}$ high-/low-level width	t_{KH2}, t_{KL2}			t_{KCY2}/2		ns
SIp setup time (to $\overline{\text{SCKp}}$↑)Note 1	t_{SIK2}			80		ns
SIp hold time (from $\overline{\text{SCKp}}$↑)Note 2	t_{KSI2}			1/f_{MCK}+50		ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp outputNote 3	t_{KSO2}	C_b = 30 pFNote 4	4.0 V ≤ V_{DD} ≤ V_b, R_b = 1.1 kΩ		2/f_{MCK}+85	ns
			V_{DD} ≤ V_b < 4.0 V, R_b = 1.8 kΩ		2/f_{MCK}+130	ns
$\overline{\text{SSIp}}$ setup time	t_{SSIK}	DAP = 0		120		ns
		DAP = 1		1/f_{MCK}+120		ns
$\overline{\text{SSIp}}$ hold time	t_{KSSI}	DAP = 0		1/f_{MCK}+120		ns
		DAP = 1		120		ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{\text{SCKp}}$↓" 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 $\overline{\text{SCKp}}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

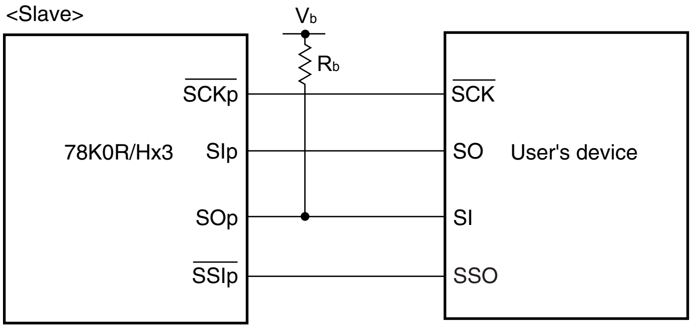
3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{\text{SCKp}}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SOp output lines.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (10/27)

CSI mode connection diagram (communication at same potential by CSI00 (when using P60 to P63))



Cautions 1. Select the normal input buffer mode for SIp and $\overline{SCKp}$ by using the PIM6 register. SOp is fixed to the N-ch open drain output (6 V tolerance) mode.

2. The timing diagram showing data transfer between pins with the same potential via CSI00 (when using P60 to P63) is the same as the diagram showing CSI transfer between pins with a different potential.

Remarks 1. p: CSI number (p = 00)

2. $R_b[\Omega]$: Communication line (SOp) pull-up resistance,
$C_b[F]$: Communication line (SOp) load capacitance, $V_b[V]$: Communication line voltage

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS00 bit of the SMR00 register.)

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (11/27)

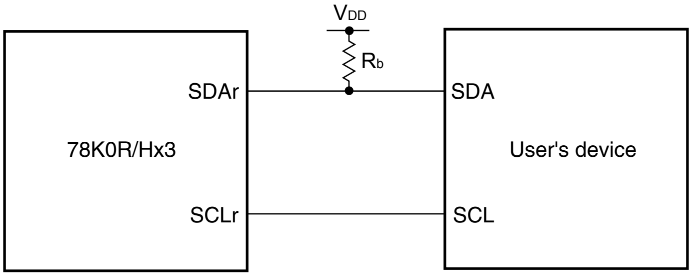
(h) During communication at same potentia (simplified I²C mode) (when IIC20 is used. SDA20 is the N-ch open drain output (V_{DD} tolerance) mode, SCL20 is the normal output mode.)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

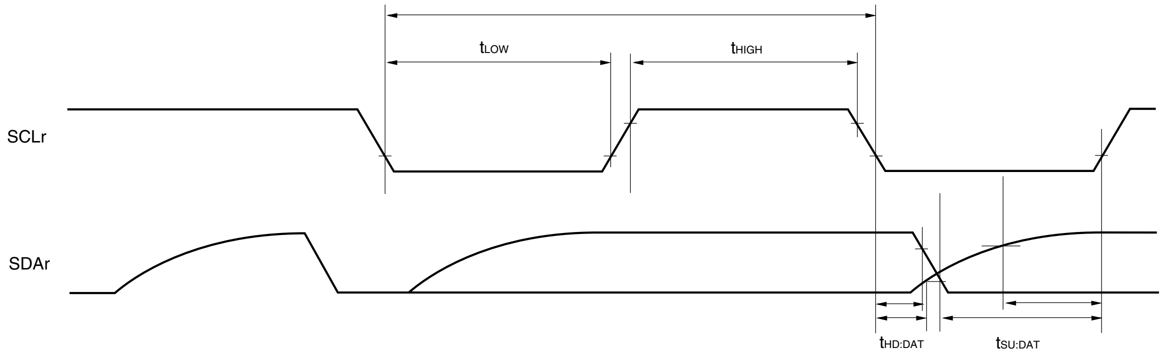
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	4.0 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ		400^Note	kHz
		2.7 V ≤ V_{DD} < 4.0 V, C_b = 100 pF, R_b = 3.0 kΩ		400^Note	kHz
Hold time when SCLr = “L”	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	995		ns
		2.7 V ≤ V_{DD} < 4.0 V, C_b = 100 pF, R_b = 3.0 kΩ	995		ns
Hold time when SCLr = “H”	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	995		ns
		2.7 V ≤ V_{DD} < 4.0 V, C_b = 100 pF, R_b = 3.0 kΩ	995		ns
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V	1/f_{MCK}+85		ns
		2.7 V ≤ V_{DD} < 4.0 V	1/f_{MCK}+120		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V < V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	0	130	ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, C_b = 100 pF, R_b = 3.0 kΩ	0	160	ns

Note And f_{SCL} ≤ f_{MCK}/4.

Simplified I²C mode mode connection diagram (during communication at same potential)



Simplified I^{2}C mode serial transfer timing (during communication at same potential)



Caution Select the N-ch open drain output (VDD tolerance) mode for SDA20 and the normal output mode for SCL20 by using the POM4 register.

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)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS20 bit of the SMR20 register)

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (12/27)

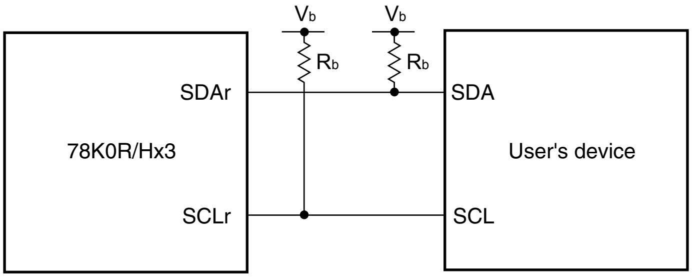
(i) During communication at same potentia (simplified I^2C mode) (when IIC20 is used. SDA20 and SCL20 are the N-ch open drain output (V_{DD} tolerance) mode.)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}			400[Note]	kHz
Hold time when SCLr = “L”	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	1300		ns
		2.7 V ≤ V_{DD} < 4.0 V, C_b = 100 pF, R_b = 5.7 kΩ			ns
Hold time when SCLr = “H”	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	600		ns
		2.7 V ≤ V_{DD} < 4.0 V, C_b = 100 pF, R_b = 5.7 kΩ			ns
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	1/f_{MCK}+120		ns
		2.7 V ≤ V_{DD} < 4.0 V, C_b = 100 pF, R_b = 5.7 kΩ	1/f_{MCK}+270		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	0	300	ns
		2.7 V ≤ V_{DD} < 4.0 V, C_b = 100 pF, R_b = 5.7 kΩ			ns

Note And f_{SCL} ≤ f_{MCK}/4.

Simplified I^2C mode connection diagram (during communication at same potential by IIC20)



Caution Select the N-ch open drain output (V_{DD} tolerance) mode for SDAr and SCLr by using the POM4 register.

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

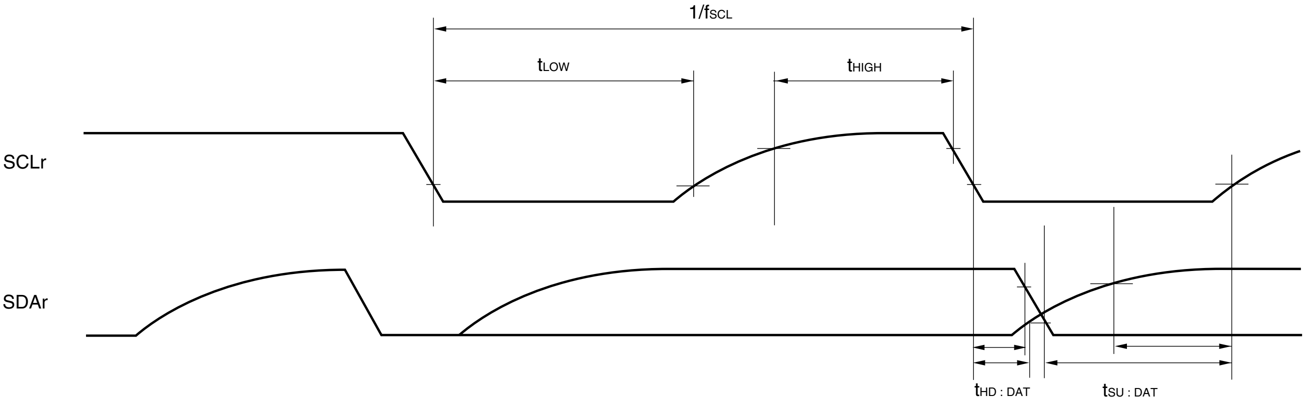
2. r: IIC number (r = 20)

3. f_{MCK}: Serial array unit operation clock frequency (Operation clock to be set by the CKS20 bit of the SMR20 register)

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (13/27)

Simplified I²C mode serial transfer timing (during communication at same potential)



Remark r: IIC number (r = 20)

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (14/27)

(j) During communication at same potentia (simplified I^{2}C mode) (when IIC11 is used. SDA11 is the normal input buffer mode and N-ch open drain output (6 V tolerance) mode, SCL11 is the N-ch open drain output (6 V tolerance) mode.)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EVDD = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EVSS = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

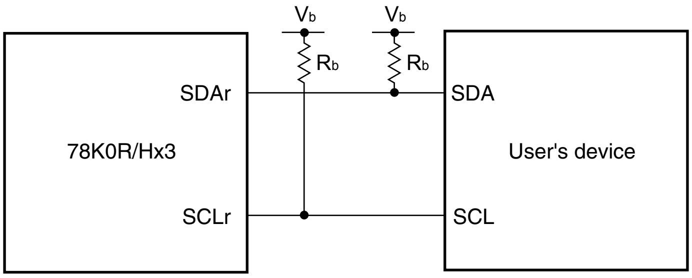
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	4.0 V ≤ V_{DD} ≤ 5.5 V, V_{DD} ≤ V_b ≤ 5.5 V, C_b = 100 pF, R_b = 1.8 kΩ		400^{Note}	kHz
		2.7 V ≤ V_{DD} < 4.0 V, V_{DD} ≤ V_b < 4.0 V, C_b = 100 pF, R_b = 1.8 kΩ			
Hold time when SCLr = “L”	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, V_{DD} ≤ V_b ≤ 5.5 V, C_b = 100 pF, R_b = 1.8 kΩ	1300		ns
		2.7 V ≤ V_{DD} < 4.0 V, V_{DD} ≤ V_b < 4.0 V, C_b = 100 pF, R_b = 1.8 kΩ			ns
Hold time when SCLr = “H”	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, V_{DD} ≤ V_b ≤ 5.5 V, C_b = 100 pF, R_b = 1.8 kΩ	600		ns
		2.7 V ≤ V_{DD} < 4.0 V, V_{DD} ≤ V_b < 4.0 V, C_b = 100 pF, R_b = 1.8 kΩ			ns
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, V_{DD} ≤ V_b ≤ 5.5 V, C_b = 100 pF, R_b = 1.8 kΩ	$1/f_{MCK}+120$		ns
		2.7 V ≤ V_{DD} < 4.0 V, V_{DD} ≤ V_b < 4.0 V, C_b = 100 pF, R_b = 1.8 kΩ	$1/f_{MCK}+135$		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, V_{DD} ≤ V_b ≤ 5.5 V, C_b = 100 pF, R_b = 1.8 kΩ	0	300	ns
		2.7 V ≤ V_{DD} < 4.0 V, V_{DD} ≤ V_b < 4.0 V, C_b = 100 pF, R_b = 1.8 kΩ			

Note And $f_{SCL} \le f_{MCK}/4$.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (15/27)

Simplified I^{2}C mode connection diagram (during communication at same potential by IIC11)



Cautions 1. Select the normal input buffer mode for SDA11 by using the PIM6 register. SCL11 and SDA11 are fixed to the N-ch open drain output (6 V tolerance) mode.

2. The timing diagram showing data transfer between pins with the same potential via IIC11 is the same as the diagram showing simplified I^{2}C transfer between pins with a different potential.

Remarks 1. R_b[Ω]: Communication line (SDAr, SCLr) pull-up resistance,
C_b[F]: Communication line (SDAr, SCLr) load capacitance, V_b[V]: Communication line voltage

2. r: IIC number (r = 11)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS11 bit of the SMR11 register)

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

<R> **(2) Serial interface: Serial array unit (16/27)**

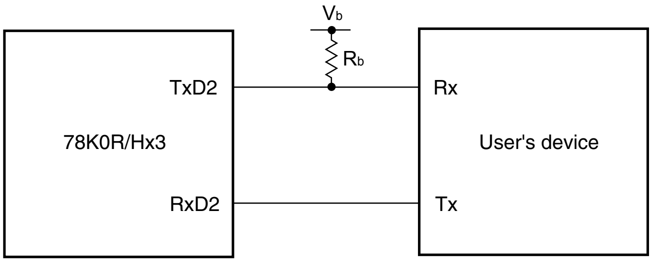
(K) Communication at different potential (UART mode) (TxD output buffer = Nch-OD, RxD input buffer = TTL)

(T_A = −40 to +140°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

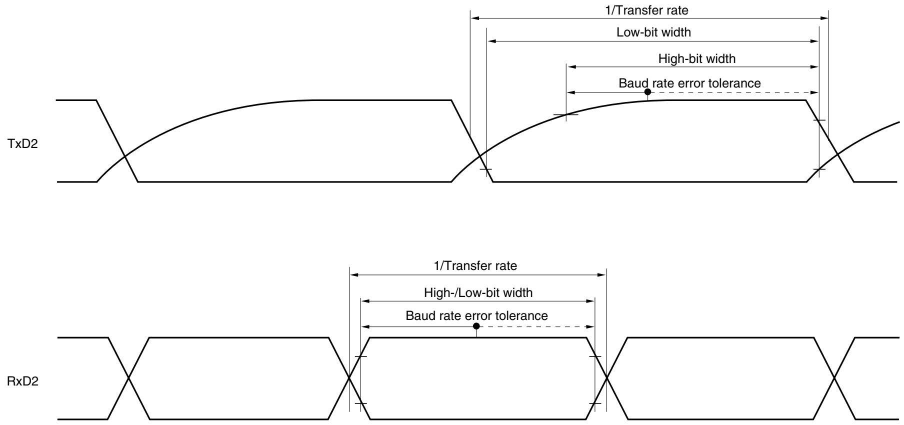
Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Transfer rate		Reception	2.7 V ≤ Vb ≤ 4.0 V V_{IH} = 2.2 V, V_{IL} = 0.8 V				f_{MCK}/6	bps
				Theoretical value of maximum Transfer rate[Note] (Cb = 30 pF)			4	Mbps
		Transmission	2.7 V ≤ Vb ≤ 4.0 V V_{OH} = 2.2 V, V_{OL} = 0.8 V				Smaller of f_{MCK}/6 and formula 1	bps
				Theoretical value of maximum transfer rate[Note] (Cb = 30 pF, Rb = 1.4 kΩ) Slew rate: Normal mode			4	Mbps

Formula 1: Maximum transfer rate = 1 / [{−Cb×Rb×ln(1−2.2/Vb)} × 3]

UART mode connection diagram (Communication at different potential)



UART mode bit width (Communication at different potential) (reference)



Note Theoretical value of maximum transfer rate is reference value to show performance of this device with a concrete value.

Remarks 1. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS2n bit of the SMR2n register. n: Channel number (n = 0, 1))

2. R_b[Ω]: Communication line (TxD) pull-up resistance,
Cb[F]: Communication line (TxD) load capacitance, Vb[V]: Communication line voltage

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (17/27)

(l) Communication at different potential (3 V) (CSI mode) (master mode, SCKp... internal clock output, slew rate: normal mode)

(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

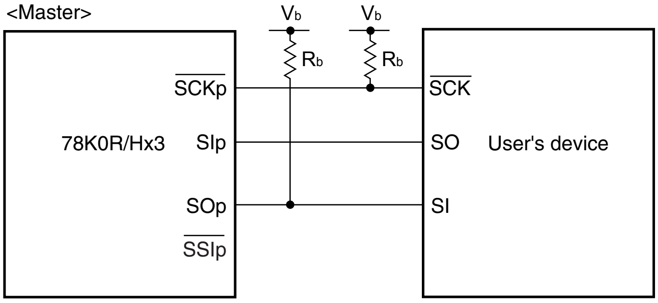
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCKp cycle time	t_{KCY1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	400 Note 3		ns
SCKp high-level width	t_{KH1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}/2–75		ns
SCKp low-level width	t_{KL1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}/2–20		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	150		ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	70		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	30		ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	30		ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ		120	ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ		40	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. And $t_{KCY1} \geq 4/t_{CLK}$.

CSI mode connection diagram (communication at different potential)



Cautions 1. For CSI00, this is the value when P15 to P17 is selected. (For details, see 11.3 (15) Serial communication pin select register (STSEL).)

2. Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and SCKp by using the PIMg and POM7 registers.

Remarks 1. p: CSI number (p = 00, 01), g: PIM and POM number (g = 6, 7)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

3. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance,
C_b[F]: Communication line (SOp, SCKp) load capacitance, V_b[V]: Communication line voltage

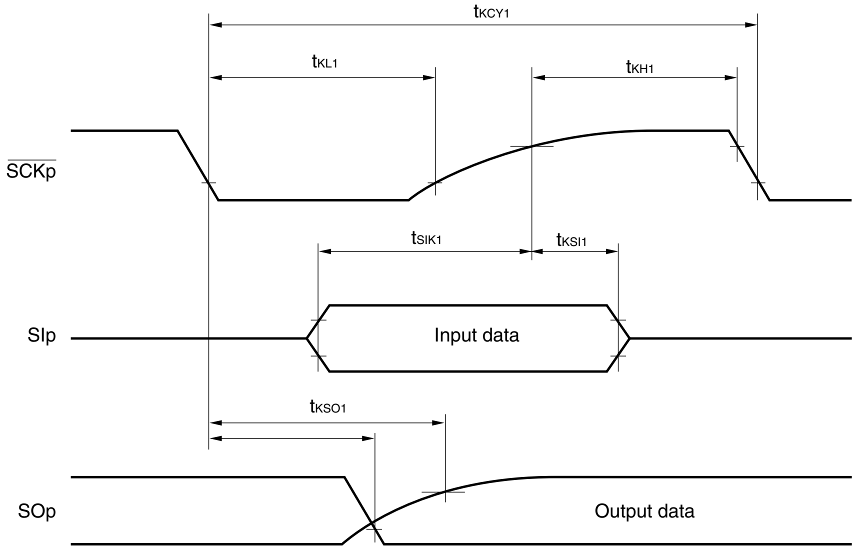
4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V

5. CSI10 and CSI11 cannot communicate at different potential. Use CSI00 and CSI01 for communication at different potential.

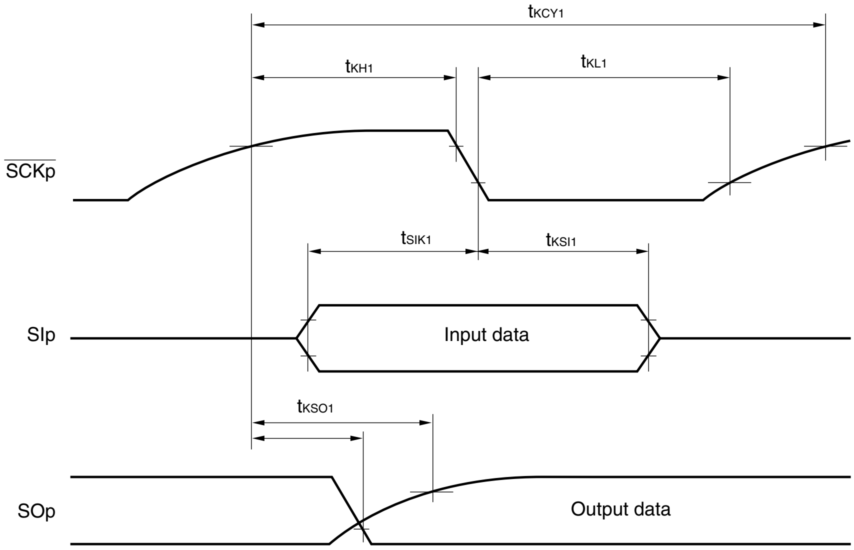
Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (18/27)

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 $\overline{\text{SCKp}}$ by using the PIMg and POM7 registers.

Remarks 1. p: CSI number (p = 00, 01), g: PIM and POM number (g = 6, 7)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

3. CSI10 and CSI11 cannot communicate at different potential. Use CSI00 and CSI01 for communication at different potential.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (19/27)

(m) Communication at different potential (3 V) (CSI mode) (slave mode, $\overline{\text{SCKp}}$... external clock input, slew rate: normal mode)

(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t_{KCY2}	2.7 V ≤ V_b ≤ V_{DD}	20 MHz < f_{MCK} ≤ 24 MHz	12/f_{MCK}		ns
			13.6 MHz < f_{MCK} ≤ 20 MHz	10/f_{MCK}		ns
			6.8 MHz < f_{MCK} ≤ 13.6 MHz	8/f_{MCK}		ns
			f_{MCK} ≤ 6.8 MHz	6/f_{MCK}		ns
$\overline{\text{SCKp}}$ high-/low-level width	t_{KH2},t_{KL2}	2.7 V ≤ V_b ≤ V_{DD}		t_{KCY2}/2–20		ns
SIp setup time (to $\overline{\text{SCKp}}$↑)[Note 1]	t_{SIK2}			90		ns
SIp hold time (from $\overline{\text{SCKp}}$↑)[Note 2]	t_{KSI2}			1/f_{MCK}+50		ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp output[Note 3]	t_{KSO2}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ			2/f_{MCK}+120	ns
$\overline{\text{SSIp}}$ setup time	t_{SSIK}	DAP = 0		120		ns
		DAP = 1		1/f_{MCK}+120		ns
$\overline{\text{SSIp}}$ hold time	t_{KSSI}	DAP = 0		1/f_{MCK}+120		ns
		DAP = 1		120		ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{\text{SCKp}}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

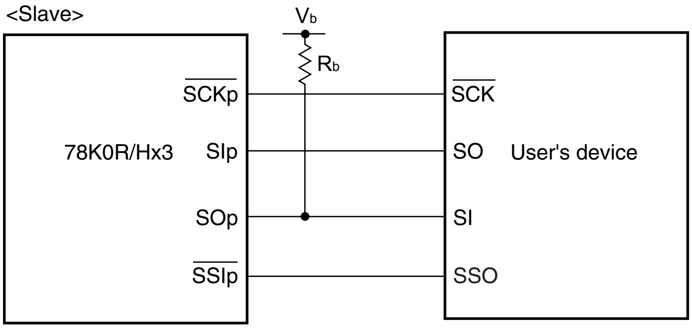
2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{\text{SCKp}}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{\text{SCKp}}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (20/27)

CSI mode connection diagram (communication at different potential)



Cautions 1. For CSI00, this is the value when P15 to P17 and 30 is selected. (For details, see 11.3 (15) Serial communication pin select register (STSEL).)

2. 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 POM7 registers.

Remarks 1. p: CSI number (p = 00, 01), g: PIM and POM number (g = 6, 7)

2. $R_b[\Omega]$: Communication line (SOp) pull-up resistance,
C_b[F]: Communication line (SOp) 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 = 0, 1), n: Channel number (n = 0, 1))

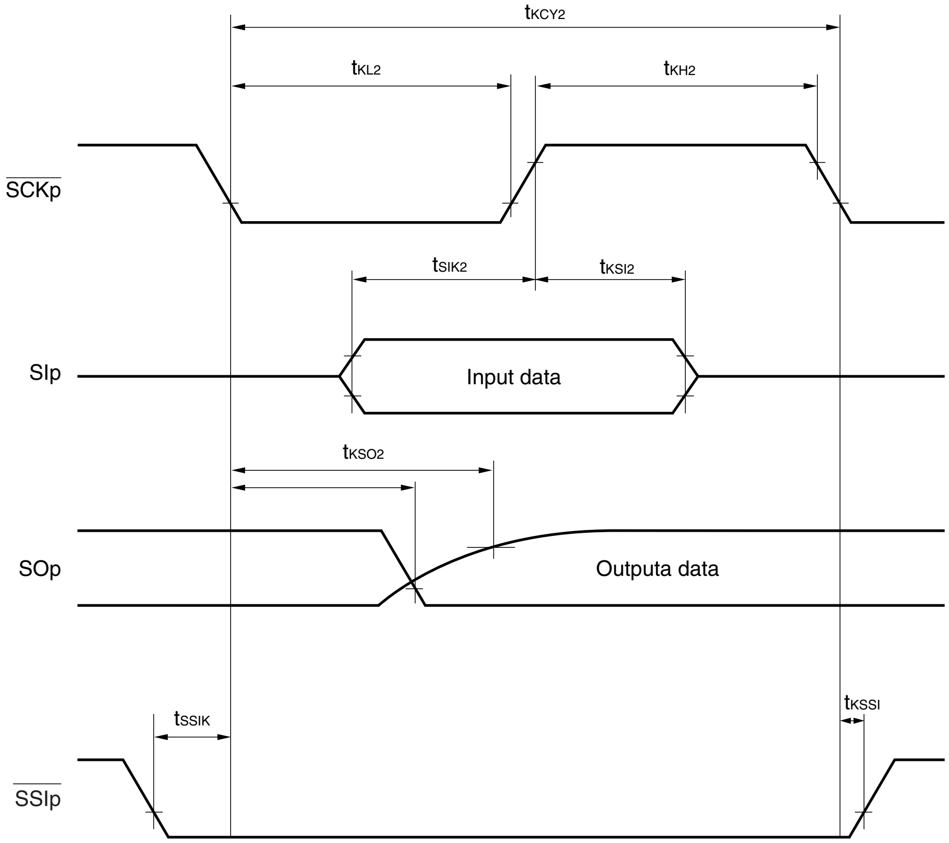
4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.
$4.0 V \leq V_{DD} \leq 5.5 V$, $2.7 V \leq V_b \leq 4.0 V$: $V_{IH} = 2.2 V$, $V_{IL} = 0.8 V$

5. CSI10 and CSI11 cannot communicate at different potential. Use CSI00 and CSI01 for communication at different potential.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (21/27)

CSI mode serial transfer timing (communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



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 POM7 registers.

Remarks 1. p: CSI number (p = 00, 01), g: PIM and POM number (g = 6, 7)

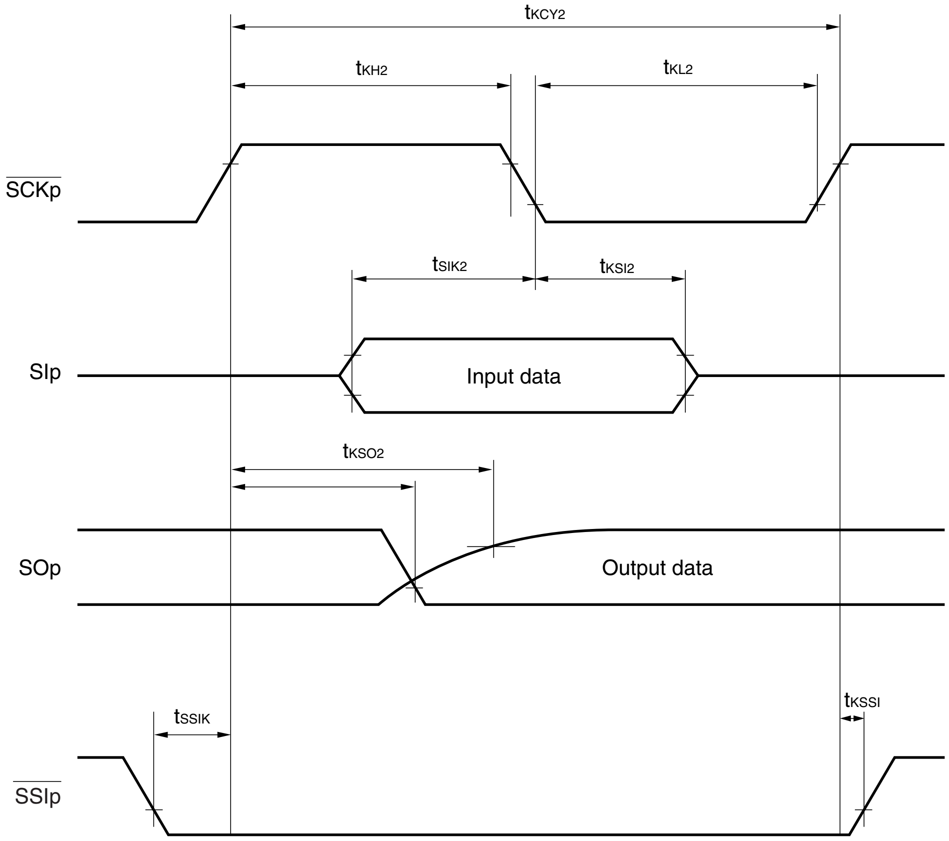
2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

3. CSI10 and CSI11 cannot communicate at different potential. Use CSI00 and CSI01 for communication at different potential

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (22/27)

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 POM7 registers.

Remarks 1. p: CSI number (p = 00, 01), g: PIM and POM number (g = 6, 7)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

3. CSI10 and CSI11 cannot communicate at different potential. Use CSI00 and CSI01 for communication at different potential.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (23/27)

(n) Communication at different potential by CSI00 (when using P60 to P63) (CSI mode) (master mode, $\overline{SCKp}$... internal clock output, slew rate: normal mode)

(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

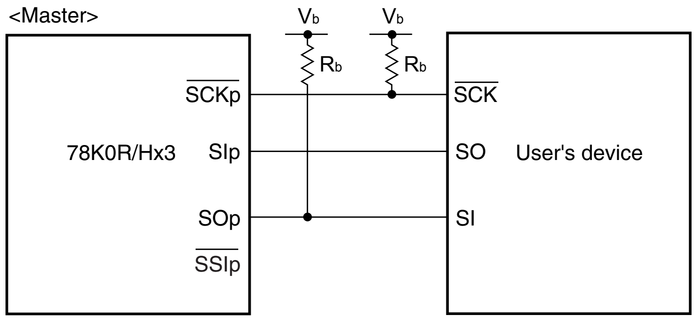
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY1}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 0.8 kΩ	400^Note 3^		ns
$\overline{SCKp}$ high-level width	t_{KH1}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 0.8 kΩ	t_{KCY1}/2–40		ns
$\overline{SCKp}$ low-level width	t_{KL1}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 0.8 kΩ	t_{KCY1}/2–20		ns
SIp setup time (to $\overline{SCKp}$↑)^Note 1^	t_{SIK1}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 0.8 kΩ	115		ns
SIp setup time (to $\overline{SCKp}$↓)^Note 2^	t_{SIK1}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 0.8 kΩ	70		ns
SIp hold time (from $\overline{SCKp}$↑)^Note 1^	t_{KSI1}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 0.8 kΩ	30		ns
SIp hold time (from $\overline{SCKp}$↓)^Note 2^	t_{KSI1}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 0.8 kΩ	30		ns
Delay time from $\overline{SCKp}$↓ to SOp output^Note 1^	t_{KSO1}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 0.8 kΩ		85	ns
Delay time from $\overline{SCKp}$↑ to SOp output^Note 2^	t_{KSO1}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 0.8 kΩ		40	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. And t_{KCY1} ≥ 4/t_{CLK}.

CSI mode connection diagram (communication at different potential by CSI00 (when using P60 to P63))



Caution Select the TTL input buffer mode for SIp by using the PIM6 register. SOp and $\overline{SCKp}$ are fixed to the N-ch open drain output (6 V tolerance) mode.

Remarks 1. p: CSI number (p = 00)

2. m: Unit number (m = 0), n: Channel number (n = 0)

3. R_b[Ω]: Communication line ($\overline{SCKp}$, SOp) pull-up resistance,
C_b[F]: Communication line (SOp, $\overline{SCKp}$) load capacitance, V_b[V]: Communication line voltage

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (24/27)

(o) Communication at different potential by CSI00 (when using P60 to P63d) (CSI mode) (slave mode, $\overline{\text{SCK}}$p... external clock input, slew rate: normal mode)

(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	MAX.	Unit
$\overline{\text{SCK}}$p cycle time	t_{KCY2}	2.7 V ≤ V_b ≤ 4.0 V	20 MHz < f_{MCK} ≤ 24 MHz	12/f_{MCK}		ns
			13.6 MHz < f_{MCK} ≤ 20 MHz	10/f_{MCK}		
			6.8 MHz < f_{MCK} ≤ 13.6 MHz	8/f_{MCK}		
			f_{MCK} ≤ 6.8 MHz	6/f_{MCK}		
$\overline{\text{SCK}}$p high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2–20		ns
SIp setup time (to $\overline{\text{SCK}}$p↑)$^{\text{Note 1}}$	t_{SIK2}			90		ns
SIp hold time (from $\overline{\text{SCK}}$p↑)$^{\text{Note 2}}$	t_{KSI2}			1/f_{MCK}+50		ns
Delay time from $\overline{\text{SCK}}$p↓ to SOp output$^{\text{Note 3}}$	t_{KSO2}	C_b = 30 pF, 2.7 V ≤ V_b ≤ 4.0 V, R_b = 0.8 kΩ			2/f_{MCK}+120	ns
$\overline{\text{SSI}}$p setup time	t_{SSIK}	DAP = 0		120		ns
		DAP = 1		1/f_{MCK}+120		ns
$\overline{\text{SSI}}$p hold time	t_{KSSI}	DAP = 0		1/f_{MCK}+120		ns
		DAP = 1		120		ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{\text{SCK}}$p↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

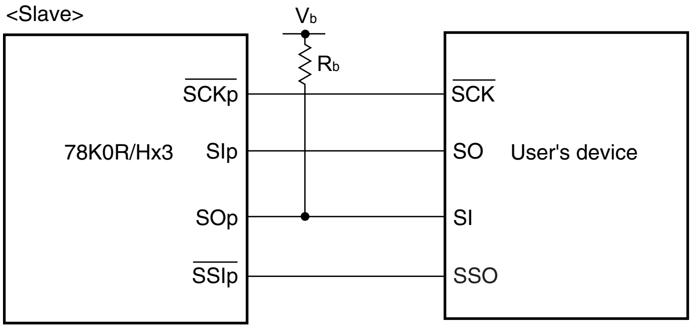
2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{\text{SCK}}$p↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{\text{SCK}}$p↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (25/27)

CSI mode connection diagram (communication at different potential by CSI00 (when using P60 to P63))



Caution Select the normal input buffer mode for SIp and $\overline{\text{SCKp}}$ by using the PIM6 register. SOp is fixed to the N-ch open drain output (6 V tolerance) mode.

Remarks 1. p: CSI number (p = 00)

2. R_b[Ω]: Communication line (SOp) pull-up resistance,
C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS00 bit of the SMR00 register.)

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (26/27)

(p) Communication at different potential (3 V) (simplified I^{2}C mode)

(when IIC11 is used. SDA11 is the TTL input buffer mode and N-ch open drain output (6 V tolerance) mode, SCL11 is the N-ch open drain output (6 V tolerance) mode.)

(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCL11 clock frequency	f_{SCL}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ		400^{Note}	kHz
Hold time when SCL11 = “L”	t_{LOW}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	1200		ns
Hold time when SCL11 = “H”	t_{HIGH}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	600		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	135+1/f_{MCK}		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	0	140	ns

Note And f_{SCL} ≤ f_{MCK}/4.

Caution Select the TTL input buffer mode for SDA11 by using the PIM6 register. SCL11 and SDA11 are fixed to the N-ch open drain output (6 V tolerance) mode.

Remarks 1. R_b[Ω]: Communication line (SDA11, SCL11) pull-up resistance,
C_b[F]: Communication line (SDA11, SCL11) load capacitance, V_b[V]: Communication line voltage

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS11 bit of the SMR11 register.)

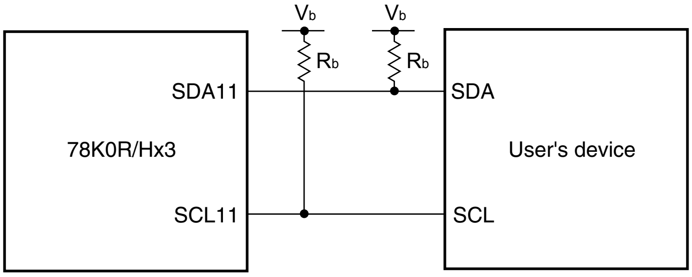
3. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in simplified I^{2}C mode mode.
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V

4. IIC20 cannot communicate at different potential. Use IIC11 for communication at different potential.

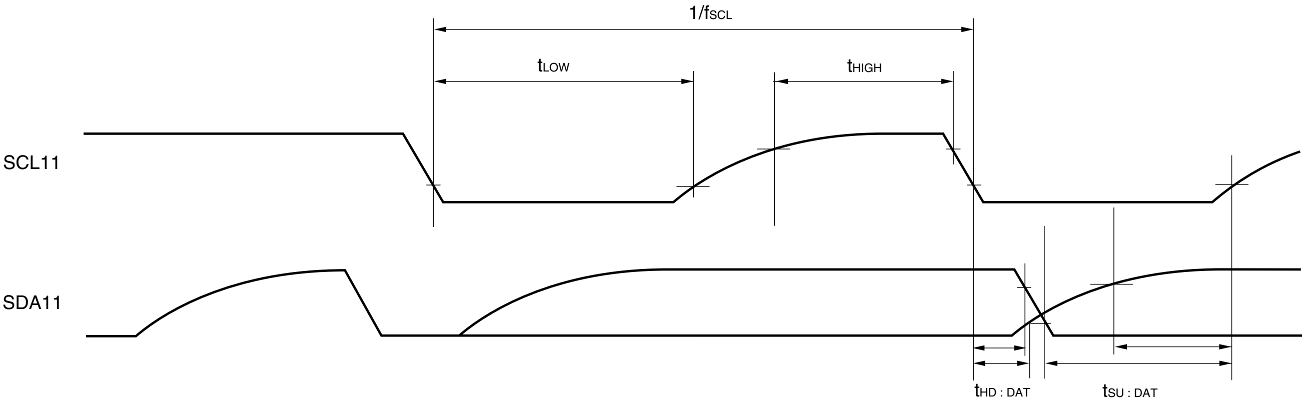
Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(2) Serial interface: Serial array unit (27/27)

Simplified I^{2}C mode connection diagram (communication at different potential)



Simplified I^{2}C mode serial transfer timing (communication at different potential)



Caution Select the TTL input buffer mode for SDA11 by using the PIM6 register. SCL11 is fixed to the N-ch open drain output (6 V tolerance) mode.

Remark R_b[Ω]: Communication line (SDA11, SCL11) pull-up resistance, V_b[V]: Communication line voltage

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(3) LIN-UART

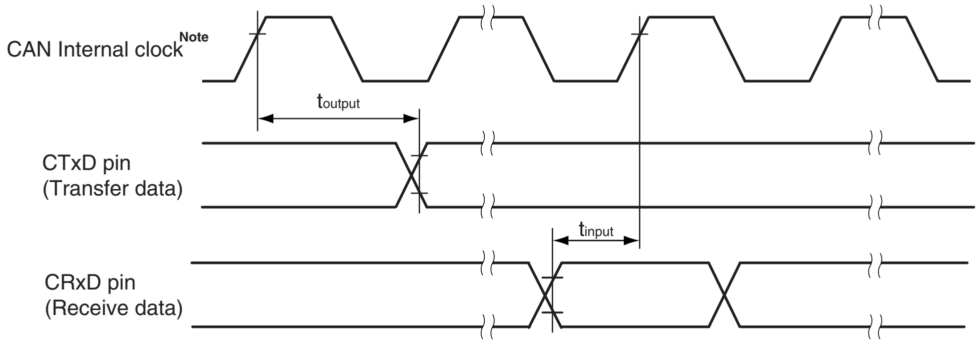
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		Slew rate: Normal mode			1	Mbps

(4) CAN controller

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					1	Mbps
Internal delay time	t_{NODE}				100	ns



Internal delay time (t_{NODE}) = Internal Transfer Delay (t_{output}) + Internal Receive Delay (t_{input})

Note CAN Internal clock (f_{CAN}): CAN baud rate clock

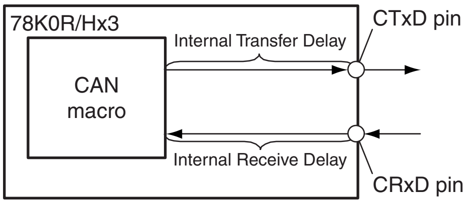


Image figure of internal delay

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

(5) Serial interface: On-chip debug (UART)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{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} ≤ 5.5 V, C_b = 30 pF)			3.33	Mbps
TOOL1 output frequency	f_{TOOL1}	C_b = 30 pF			12	MHz

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

A/D Converter Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}				10	bit
Overall error[Notes 1, 2]	AINL	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.3	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.5	%FSR
Conversion time	t_{CONV}	High speed mode 1 (4.0 V ≤ AV_{REF} ≤ 5.5 V)	2.75		77	μs
		High speed mode 2	3.9		77	μs
		Normal mode	9.3		77	μs
Zero-scale error[Notes 1, 2]	EZS	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.3	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.5	%FSR
Full-scale error[Notes 1, 2]	EFS	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.3	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.5	%FSR
Integral non-linearity error[Note 1]	ILE	4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±3.5	LSB
Differential non-linearity error[Note 1]	DLE	4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±1.5	LSB
Analog input voltage	V_{AIN}		AV_{SS}		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

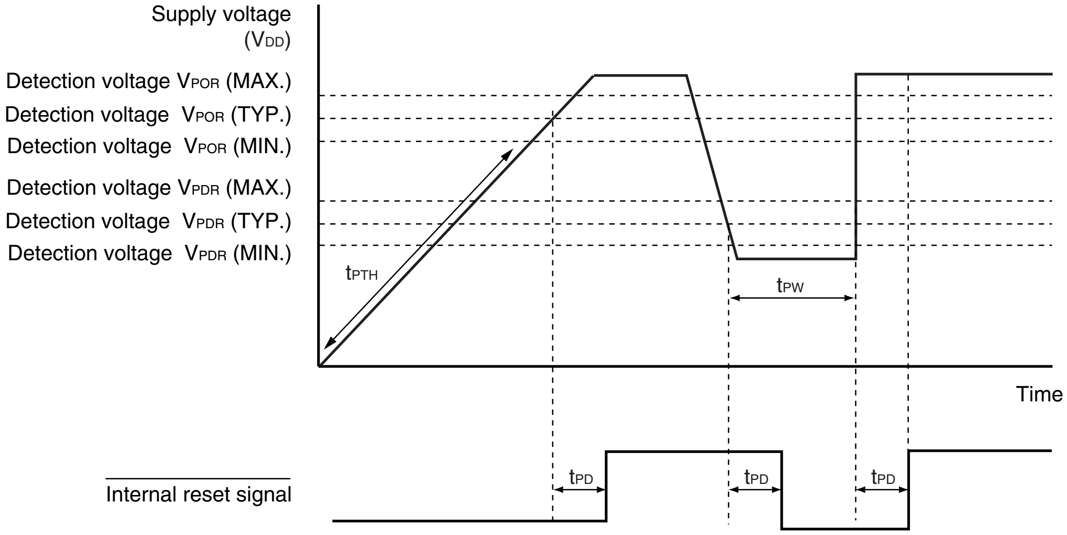
2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

POC Circuit Characteristics (T_A = −40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power on detection	1.52	1.61	1.70	V
	V_{PDR}	Power down detection	1.5	1.59	1.68	V
Power supply voltage rise inclination	t_{PTH}	Change inclination of V_{DD}: 0 V → V_{POC0}	0.5			V/ms
Minimum pulse width	t_{PW}	When the voltage drops	200			μs
Detection delay time	t_{PD}				200	μs

POC Circuit Timing


Supply voltage
(VDD)
Detection voltage VPOR (MAX.)
Detection voltage VPOR (TYP.)
Detection voltage VPOR (MIN.)
Detection voltage VPDR (MAX.)
Detection voltage VPDR (TYP.)
Detection voltage VPDR (MIN.)
tPTH
tPW
Time
Internal reset signal
tPD
tPD
tPD

Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

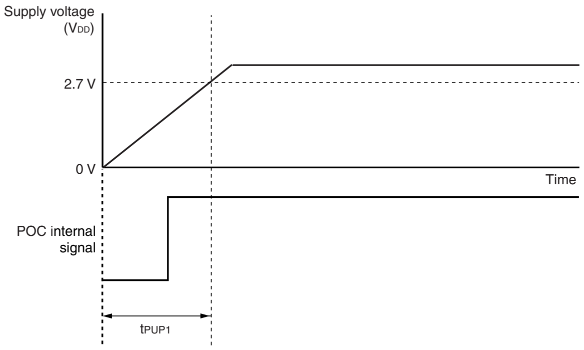
Supply Voltage Rise Time (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 2.7 V (V_{DD} (MIN.))[Note] (V_{DD}: 0 V → 2.7 V)	t_{PUP1}	LVI default dtart function stopped is set ($\overline{\text{LVIOFF}}$ (Option Byte) = 1), when $\overline{\text{RESET}}$ input is not used			5.4	ms
Maximum time to rise to 2.7 V (V_{DD} (MIN.))[Note] (releasing $\overline{\text{RESET}}$ input → V_{DD}: 2.7 V)	t_{PUP2}	LVI default dtart function stopped is set ($\overline{\text{LVIOFF}}$ (Option Byte) = 1), when $\overline{\text{RESET}}$ input is used			1.88	ms

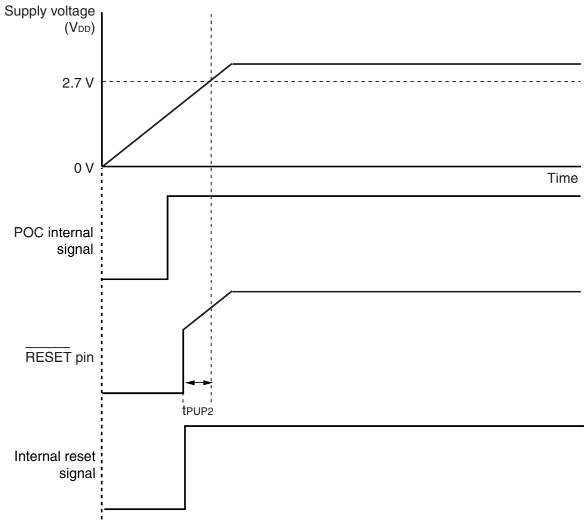
Note Make sure to raise the power supply in a shorter time than this.

Supply Voltage Rise Time Timing

- When $\overline{\text{RESET}}$ pin input is not used



- When $\overline{\text{RESET}}$ pin input is used (when external reset is released by the $\overline{\text{RESET}}$ pin, after POC has been released)



Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

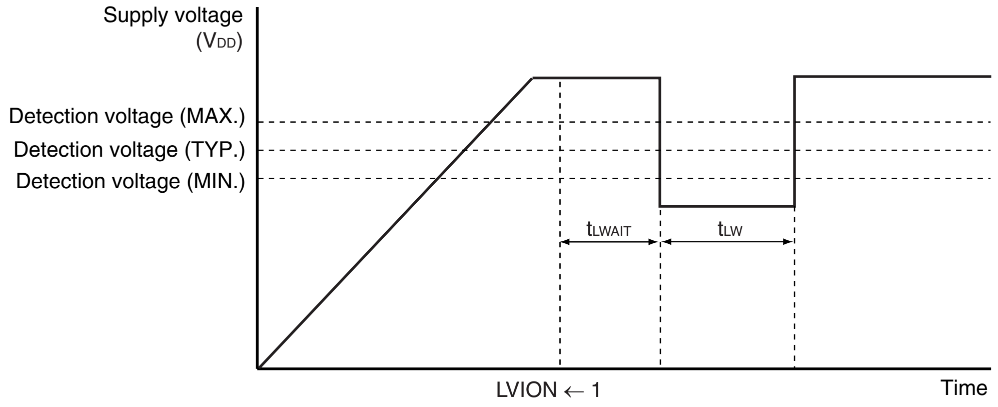
LVI Circuit Characteristics (T_A = −40 to +85°C, $V_{POC} \le V_{DD} = EV_{DD} = EV_{DD1} \le 5.5$ V, $V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = 0$ V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.12	4.22	4.32	V
		V_{LVI1}		3.97	4.07	4.17	V
		V_{LVI2}		3.82	3.92	4.02	V
		V_{LVI3}		3.66	3.76	3.86	V
		V_{LVI4}		3.51	3.61	3.71	V
		V_{LVI5}		3.35	3.45	3.55	V
		V_{LVI6}		3.20	3.30	3.40	V
		V_{LVI7}		3.05	3.15	3.25	V
		V_{LVI8}		2.89	2.99	3.09	V
		V_{LVI9}		2.74	2.84	2.94	V
	External input pin[Note]	V_{EXLVI}	EXLVI ≤ V_{DD}, 2.7 V ≤ V_{DD} ≤ 5.5 V	1.11	1.21	1.31	V
	Supply voltage when power supply voltage is turned on	V_{PUPLVI}	When LVI default start function enabled is set	2.73	2.93	3.13	V
Minimum pulse width		t_{LW}		200			μs
Detection delay time		t_{LD}				200	μs
Operation stabilization wait time		t_{LWAIT}				10	μs

Note The EXLVI/P120/INTP0 pin is used.

Remark $V_{LVI(n-1)} > V_{LVIn}$: n = 1 to 9

<R> **LVI Circuit Timing**



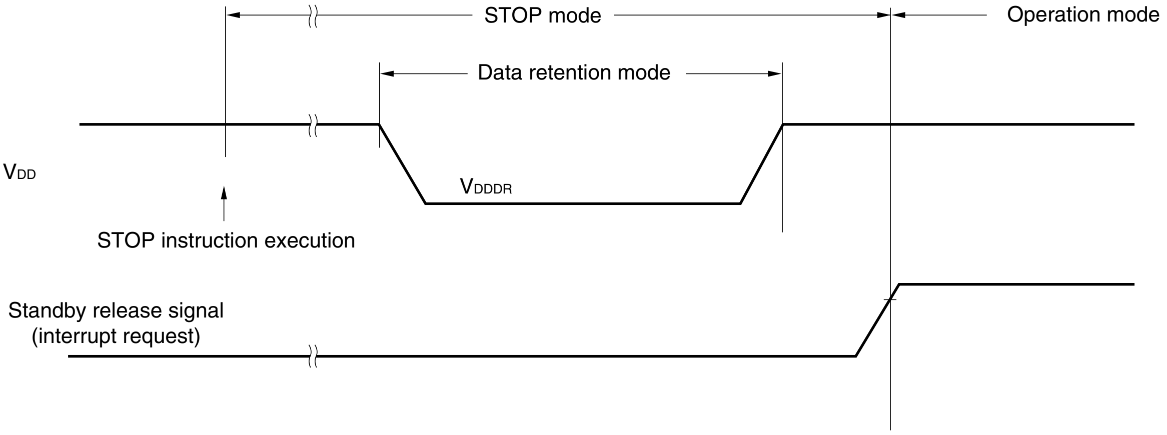
Caution The pins mounted depend on the product. See Caution 2 at the beginning of this chapter.

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[Note 1]	V_{DDDR}		1.5[Note 2]		5.5	V

Notes 1. The data when a reset is effected in a state other than the STOP state is not guaranteed.

2. 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.



Code Flash Memory Programming Characteristics
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
V_{DD} supply current[Note]	I_{DD}	Typ. = 10 MHz, Max. = 24 MHz			4.5	20	mA
CPU/peripheral hardware clock frequency	f_{CLK}			2		24	MHz
Number of rewrites (number of deletes per block)	C_{erwr}	Used for updating programs When using flash memory programmer and Renesas Electronics self programming library	Retained for 15 years	1,000			Times

Note This value includes the code flash programming current and chip operating current.

<R>

Data Flash Memory Programming Characteristics
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{DD} supply current[Note 1]	I_{DD}			6	12	mA
CPU/peripheral hardware clock frequency	f_{CLK}				24	MHz
Number of rewrites per sector	S_{erwr}	Data retained for 20 years[Note 2]	10000			Times

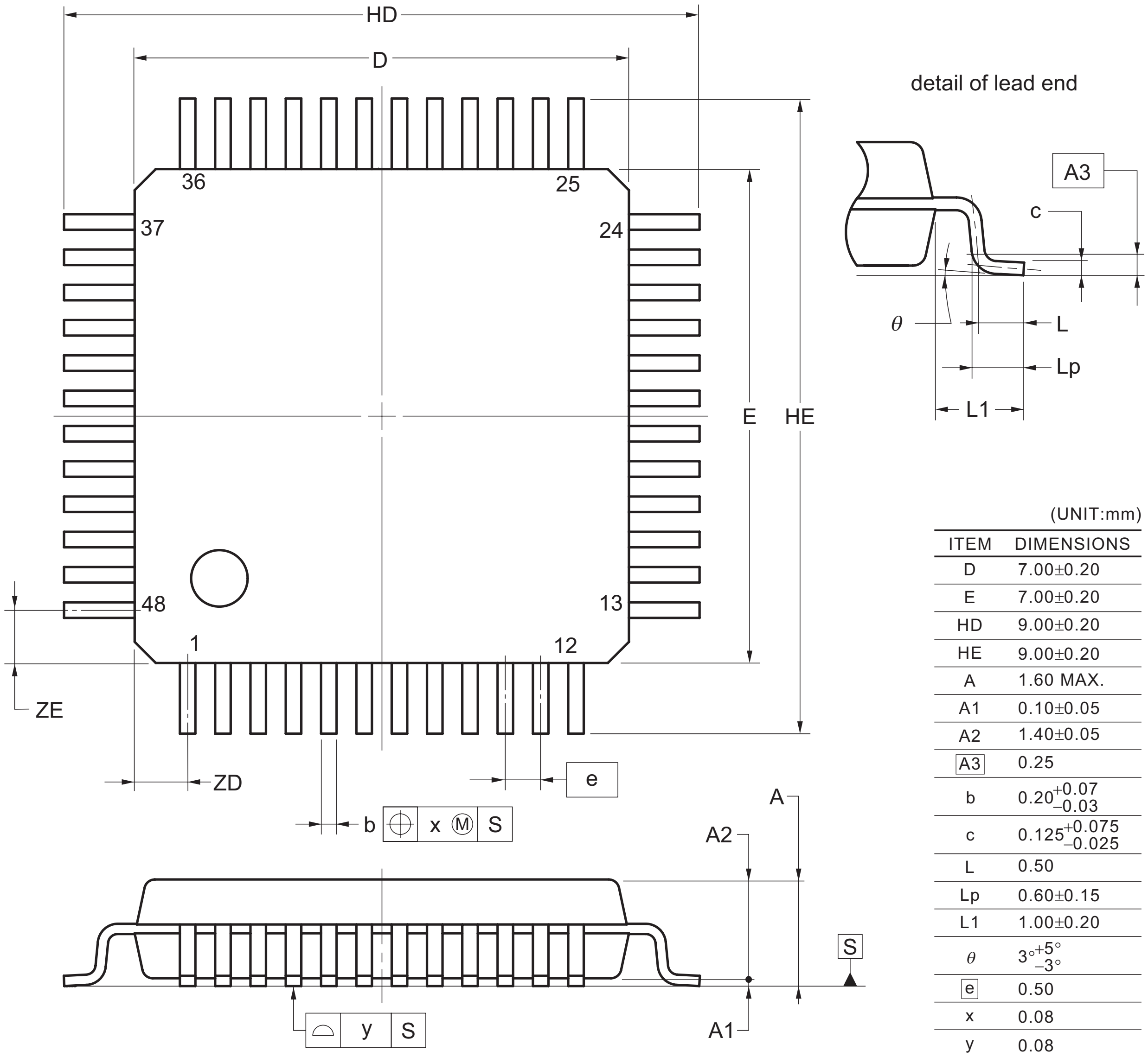
Notes 1. This is the current consumption of only the data flash.

2. This indicates the number of years the data can be retained from the time the data flash is first written to. This applies not only to the block that is first written to, but to all the blocks.

CHAPTER 30 PACKAGE DRAWINGS

30.1 78K0R/HC3

48-PIN PLASTIC LQFP (FINE PITCH) (7x7)



(UNIT:mm)

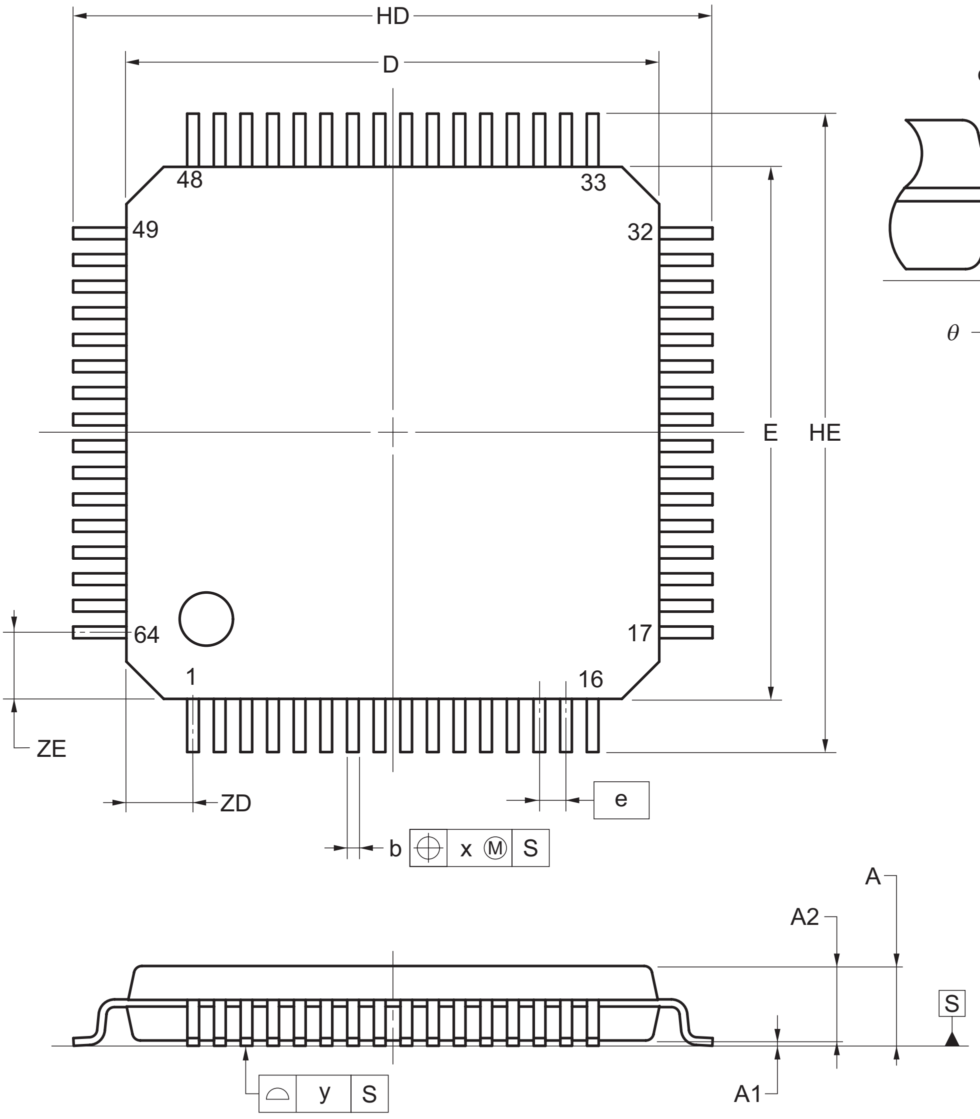
ITEM	DIMENSIONS
D	7.00±0.20
E	7.00±0.20
HD	9.00±0.20
HE	9.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3°^{+5°}_{-3°}$
e	0.50
x	0.08
y	0.08
ZD	0.75
ZE	0.75

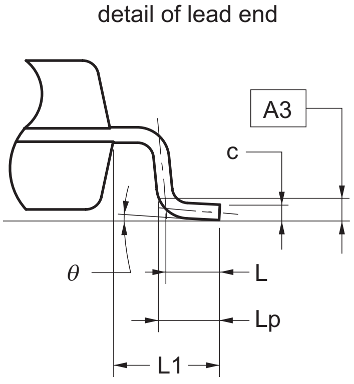
P48GA-50-GAM

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

30.2 78K0R/HE3

64-PIN PLASTIC LQFP(FINE PITCH)(10x10)





NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

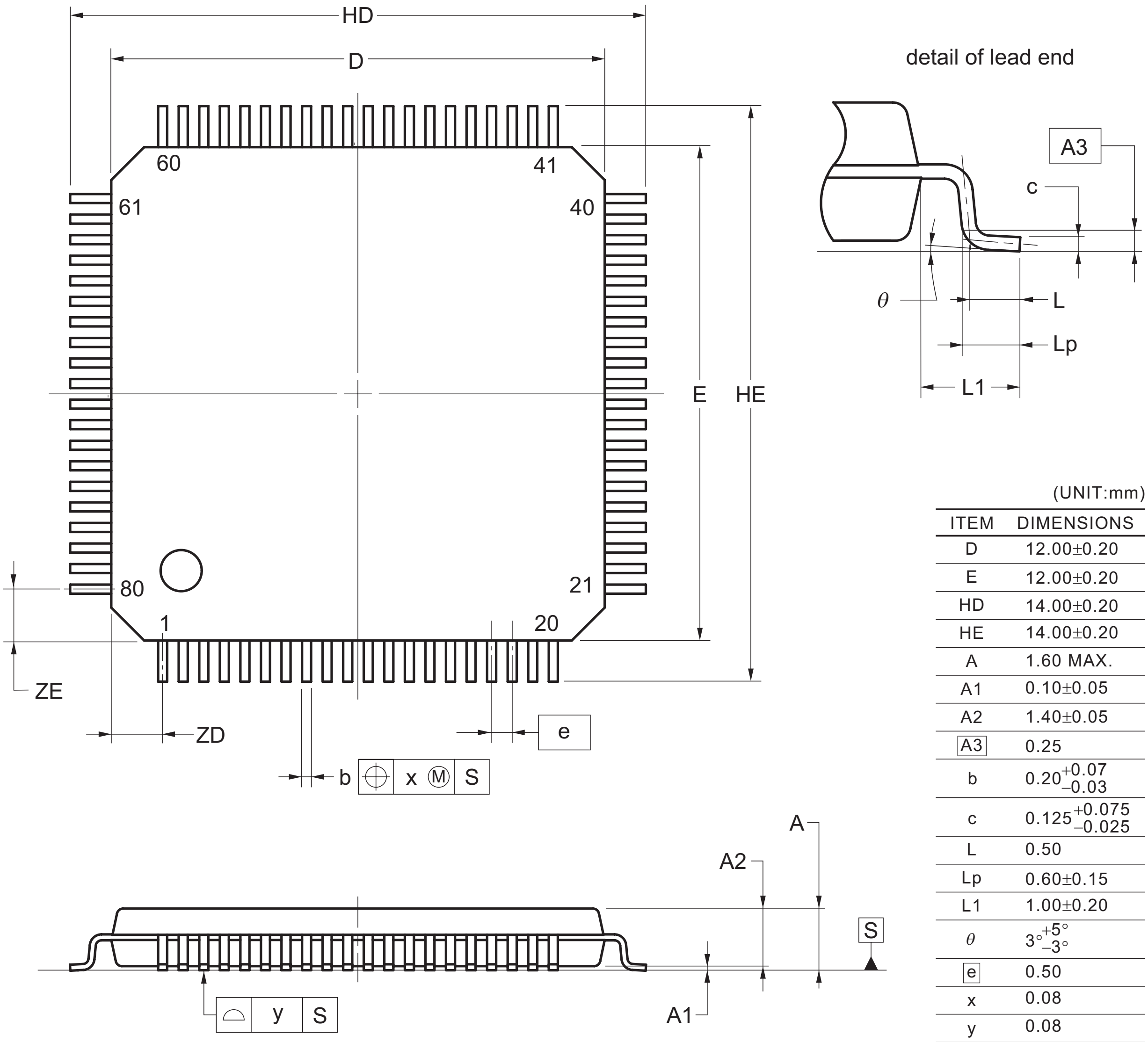
(UNIT:mm)

ITEM	DIMENSIONS
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
$\boxed{A3}$	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3°^{+5°}_{-3°}$
$\boxed{e}$	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

P64GB-50-GAH

30.3 78K0R/HF3

80-PIN PLASTIC LQFP (FINE PITCH) (12x12)



(UNIT:mm)

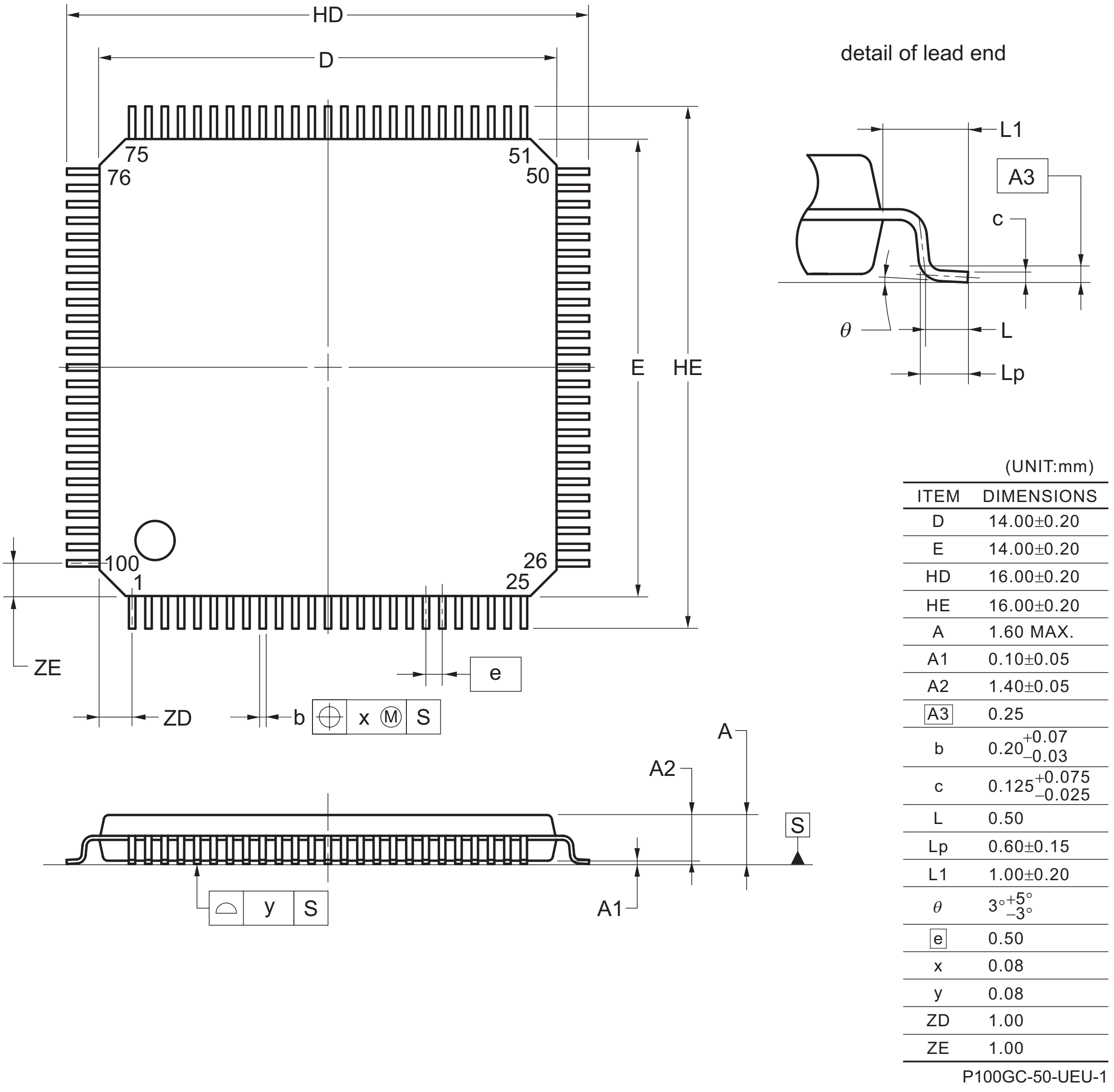
ITEM	DIMENSIONS
D	12.00±0.20
E	12.00±0.20
HD	14.00±0.20
HE	14.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
$\boxed{\text{A3}}$	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3°^{+5°}_{-3°}$
$\boxed{\text{e}}$	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

P80GK-50-GAK

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

30.4 78K0R/HG3

100-PIN PLASTIC LQFP (FINE PITCH) (14x14)



(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
$\boxed{\text{A3}}$	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3°^{+5°}_{-3°}$
$\boxed{\text{e}}$	0.50
x	0.08
y	0.08
ZD	1.00
ZE	1.00

P100GC-50-UEU-1

APPENDIX A DEVELOPMENT TOOLS

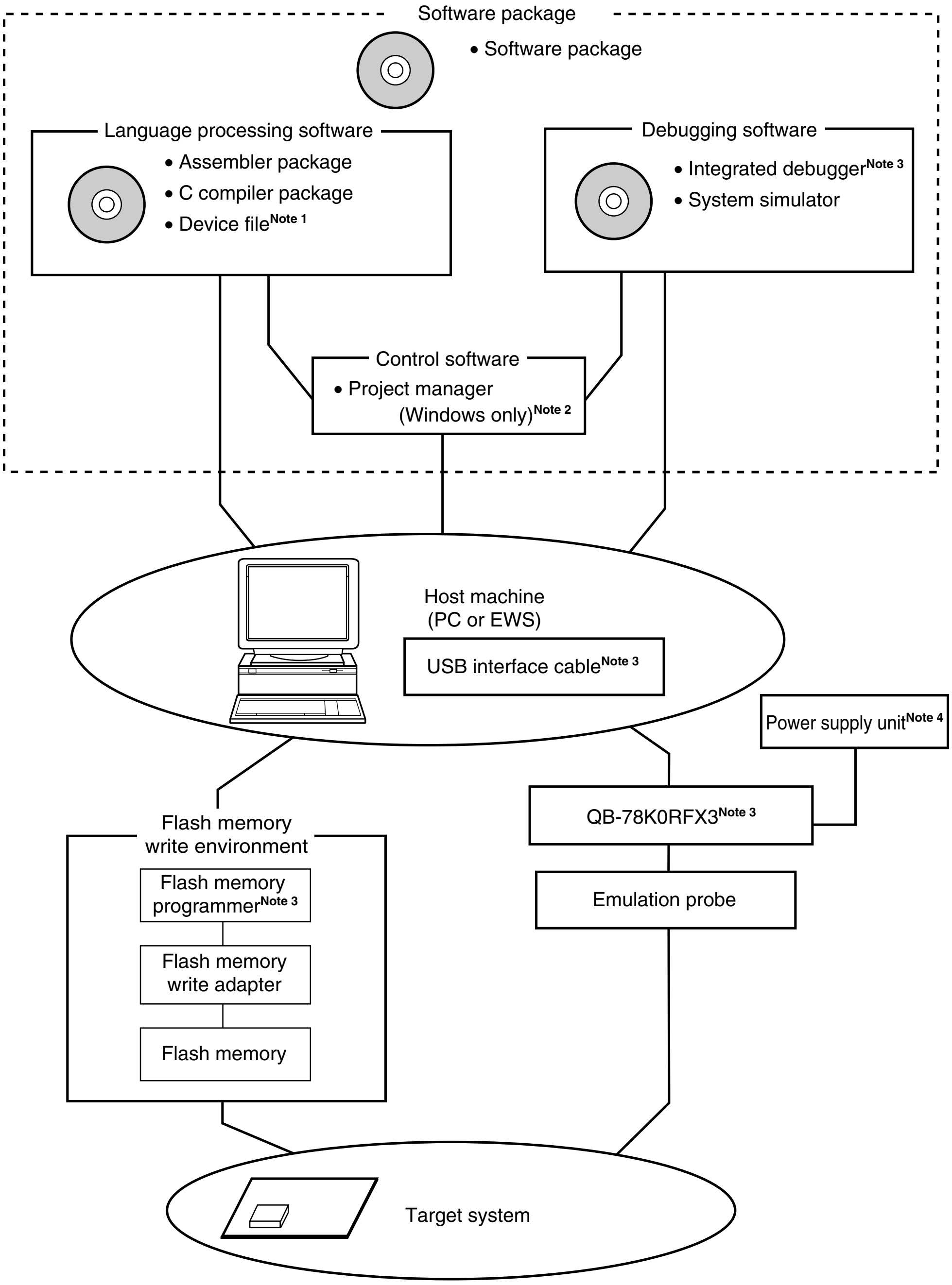
The following development tools are available for the development of systems that employ the 78K0R/Hx3.
Figure A-1 shows the development tool configuration.

- **Support for PC98-NX series**
  Unless otherwise specified, products supported by IBM PC/AT™ compatibles are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT compatibles.

- **Windows™**
  Unless otherwise specified, "Windows" means the following OSs.
  - Windows 98
  - Windows NT™
  - Windows 2000
  - Windows XP

Figure A-1. Development Tool Configuration (1/2)

(1) When using the in-circuit emulator QB-78K0RFX3



Notes 1. Download the device file for 78K0R/Hx3 (DF781050) from the download site for development tools (http://www2.renesas.com/micro/en/ods/).

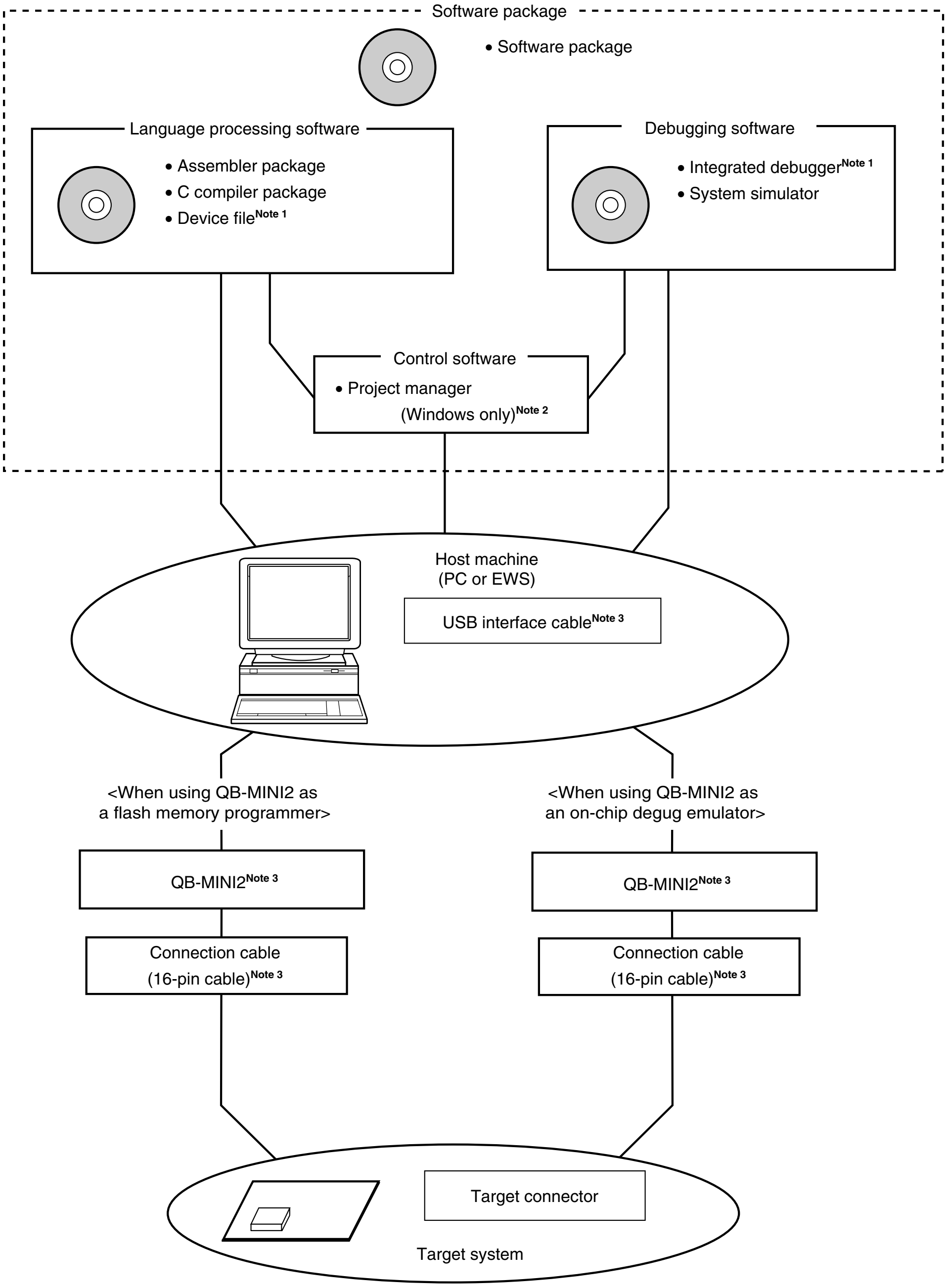
2. The project manager PM+ is included in the assembler package. The PM+ is only used for Windows.

3. In-circuit emulator QB-78K0RFX3 is supplied with integrated debugger ID78K0R-QB, on-chip debug emulator with programming function QB-MINI2, power supply unit, and USB interface cable. Any other products are sold separately.

4. The power supply unit does not include the QB-78K0RFX3. The power supply unit must be purchased separately.

Figure A-1. Development Tool Configuration (2/2)

(2) When using the on-chip debug emulator with programming function QB-MINI2



Notes 1. Download the device file for 78K0R/Hx3 (DF781050) and the integrated debugger (ID78K0R-QB) from the download site for development tools (http://www2.renesas.com/micro/en/ods/).

2. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

3. On-chip debug emulator QB-MINI2 is supplied with USB interface cable, connection cables (10-pin cable and 16-pin cable), and 78K0-OCD board. Any other products are sold separately. In addition, download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html).

A.1 Software Package

SP78K0R 78K0R Series software package	Development tools (software) common to the 78K0R microcontrollers are combined in this package.
	Part number: μS××××SP78K0R

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SP78K0R

××××	Host Machine	OS
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)
BB17		Windows (English version)

A.2 Language Processing Software

RA78K0R Assembler package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. This assembler is also provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with a device file (DF781050). **<Precaution when using RA78K0R in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: μS××××RA78K0R
CC78K0R C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an assembler package and device file (both sold separately). **<Precaution when using CC78K0R in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: μS××××CC78K0R
DF781050[Note] Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0R, CC78K0R, and ID78K0R-QB) (all sold separately). The corresponding OS and host machine differ depending on the tool to be used.
	Part number: μS××××DF781050

Note The DF781050 can be used in common with the RA78K0R, CC78K0R, and ID78K0R-QB. Download the DF781050 from the download site for development tools (http://www2.renesas.com/micro/en/ods/).

Remark ×××× in the part number differs depending on the host machine and OS used.

*μ*S××××RA78K0R
*μ*S××××CC78K0R

××××	Host Machine	OS
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)
BB17		Windows (English version)

*μ*S××××DF781050

××××	Host Machine	OS
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)
BB13		Windows (English version)

A.3 Control Software

PM+ Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from the project manager. **<Caution>** The project manager is included in the assembler package (RA78K0R). It can only be used in Windows.

A.4 Flash Memory Programming Tools

A.4.1 When using flash memory programmer FG-FP5, and FL-PR5

FL-PR5, PG-FP5 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
FA-xxxx[Note] Flash memory programming adapter	Flash memory programming adapter used connected to the flash memory programmer for use.

Note The part numbers of the flash memory programming adapter and the packages of the target device are described below.

Package		Flash Memory Programming Adapter
78K0R/HC3	48-pin plastic LQFP (GA-GAM type)	FA-78F1828GA-GAM-RX
78K0R/HE3	64-pin plastic LQFP (GB-GAH type)	FA-78F1833GB-GAH-RX
78K0R/HF3	80-pin plastic LQFP (GK-GAK type)	FA-78F1840GK-GAK-RX
78K0R/HG3	100-pin plastic LQFP (GC-UEU type)	FA-78F1845GC-UEU-RX

Remarks 1. FL-PR5, and FA-xxxx are products of Naito Densei Machida Mfg. Co., Ltd.
TEL: +81-42-750-4172 Naito Densei Machida Mfg. Co., Ltd.

2. Use the latest version of the flash memory programming adapter.

A.4.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This is a flash memory programmer dedicated to microcontrollers with on-chip flash memory. It is available also as on-chip debug emulator which serves to debug hardware and software when developing application systems using the 78K0R. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. To use 78K0R/Hx3, use USB interface cable and 16-pin connection cable.

Remark Download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html).

A.5 Debugging Tools (Hardware)

A.5.1 When using in-circuit emulator QB-78K0RFX3

QB-78K0RFX3 In-circuit emulator	This in-circuit emulator serves to debug hardware and software when developing application systems using the 78K0R/Hx3. It supports to the integrated debugger (ID78K0R-QB). This emulator should be used in combination with a power supply unit and emulation probe, and the USB is used to connect this emulator to the host machine.
QB-144-CA-01 Check pin adapter	This check pin adapter is used in waveform monitoring using the oscilloscope, etc.

QB-144-EP-02S^{Note 1}
QB-80-EP-01T^{Note 2}
Emulation probe
	This emulation probe is flexible type and used to connect the in-circuit emulator and target system.

QB-xxxx-EA-xxx^{Note 3} Exchange adapter	This exchange adapter is used to perform pin conversion from the in-circuit emulator to target connector.
QB-xxxx-YS-xxx^{Note 3} Space adapter	This space adapter is used to adjust the height between the target system and in-circuit emulator.
QB-xxxx-YQ-xxx^{Note 3} YQ connector	This YQ connector is used to connect the target connector and exchange adapter.
QB-xxxx-HQ-xxx^{Note 3} Mount adapter	This mount adapter is used to mount the target device with socket.
QB-xxxx-NQ-xxx^{Note 3} Target connector	This target connector is used to mount on the target system.

Notes 1. 78K0R/HG3 only

2. 78K0R/HC3, 78K0R/HE3, 78K0R/HF3 only

3. The part numbers of the exchange adapter, space adapter, YQ connector, mount adapter, and target connector and the packages of the target device are described below.

Package		Exchange Adapter	Space Adapter	YQ Connector	Mount Adapter	Target Connector
78K0R/HC3	48-pin plastic LQFP (GA-GAM type)	QB-48GA-EA-05T	QB-48GA-YS-01T	QB-48GA-YQ-01T	QB-48GA-HQ-01T	QB-48GA-NQ-01T
78K0R/HE3	64-pin plastic LQFP (GB-GAH type)	QB-64GB-EA-09T	QB-64GB-YS-01T	QB-64GB-YQ-01T	QB-64GB-HQ-01T	QB-64GB-NQ-01T
78K0R/HF3	80-pin plastic LQFP (GK-GAK type)	QB-80GK-EA-10T	QB-80GK-YS-01T	QB-80GK-YQ-01T	QB-80GK-HQ-01T	QB-80GK-NQ-01T
78K0R/HG3	100-pin plastic LQFP (GC-UEU type)	QB-100GC-EA-09T	QB-100GC-YS-01T	QB-100GC-YQ-01T	QB-100GC-HQ-01T	QB-100GC-NQ-01T

(**Remarks 1** to **3** are listed on the next page.)

Remarks **1.** The QB-78K0RFX3 is supplied with a power supply unit and USB interface cable. As control software, integrated debugger ID78K0R-QB and on-chip debug emulator with programming function QB-MINI2 are supplied.

2. The power supply unit does not include the QB-78K0RFX3. The power supply unit must be purchased separately.

3. The packed contents differ depending on the part number, as follows.

Packed Contents Part Number	In-Circuit Emulator	Emulation Probe	Exchange Adapter	YQ Connector	Target Connector
QB-78K0RFX3-ZZZ	QB-78K0RFX3	None			
QB-78K0RFX3-T48GA		QB-80-EP-01T	QB-48GA-EA-05T	QB-48GA-YQ-01T	QB-48GA-NQ-01T
QB-78K0RFX3-T64GB			QB-64GB-EA-09T	QB-64GB-YQ-01T	QB-64GB-NQ-01T
QB-78K0RFX3-T80GK			QB-80GK-EA-10T	QB-80GK-YQ-01T	QB-80GK-NQ-01T
QB-78K0RFX3-T100GC		QB-144-EP-02S	QB-100GC-EA-09T	QB-100GC-YQ-01T	QB-100GC-NQ-01T

A.5.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This on-chip debug emulator serves to debug hardware and software when developing application systems using the 78K0R/Hx3 microcontrollers. It is available also as flash memory programmer dedicated to microcontrollers with on-chip flash memory. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. To use the 78K0R/Hx3, use USB interface cable and 16-pin connection cable.

Remark Download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html).

A.6 Debugging Tools (Software)

ID78K0R-QB Integrated debugger	This debugger supports the in-circuit emulators for the 78K0R microcontrollers. The ID78K0R-QB is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with the device file.
	Part number: µS××××ID78K0R-QB

Remark ×××× in the part number differs depending on the host machine and OS used.

µS××××ID78K0R-QB

××××	Host Machine	OS
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)
BB17		Windows (English version)

APPENDIX B REVISION HISTORY

B.1 Major Revisions in This Edition

(1/4)

Page	Description	Classification
CHAPTER 1 OUTLINE		
p.20 to 23	Change of **1.4 Pin Configuration (Top View)**	(b)
p.29	Change of **1.7 Outline of Functions**	(a)
CHAPTER 2 PIN FUNCTIONS		
p.33 to 60	Change of **2.1 Pin Function List**	(c)
p.62 to 80	Change of **2.2 Description of Pin Functions**	(c)
p.81	Change of **2.3 Pin I/O Circuits and Recommended Connection of Unused Pins**	(c)
CHAPTER 3 CPU ARCHITECTURE		
p.96	Change of **3.1 Memory Space**	(c)
p.133	Change of **3.3 Instruction Address Addressing**	(c)
p.141 to 144	Change of **3.4 Addressing for Processing Data Addresses**	(c)
CHAPTER 4 PORT FUNCTIONS		
p.159 to 228	Change of **4.2 Port Configuration**	(c)
p.240 to 254	Change of **4.3 Registers Controlling Port Function**	(c)
p.266	Change of **4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function**	(c)
CHAPTER 5 CLOCK GENERATOR		
p.281 to 289	Change of **5.3 Registers Controlling Clock Generator**	(c)
p.310	Change of **5.7 Controlling Clock**	(c)
CHAPTER 6 TIMER ARRAY UNIT		
p.338 to 366	Change of **6.3 Registers Controlling Timer Array Unit**	(c)
p.377	Change of **6.5 Channel Input (TImn Pin) Control**	(c)
p.405	Change of **6.7 Operation of Timer Array Unit as Independent Channel**	(c)
p.407 to 434	Change of **6.8 Operation of Plural Channels of Timer Array Unit**	(c)
CHAPTER 7 16-BIT WAKEUP TIMER		
p.438, 439	Change of **7.3 Register**	(c)
p.441	Change of **7.4 Operation**	(c)
CHAPER 9 CLOCK OUTPUT CONTROLLER		
p.454, 455	Change of **9.3 Registers Controlling Clock Output Controller**	(c)
CHAPTER 10 A/D CONVERTER		
p.470 to 476	Change of **10.3 Registers Controlling A/D Converter**	(c)
p.490	Change of **10.4 A/D Converter Operations**	(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

(2/4)

Page	Description	Classification
CHAPTER 11 SERIAL ARRAY UNIT		
p.504	Change of **Table 11-2. Configuration of Serial Array Unit**	(c)
p.506	Modification of description in **Figure 11-2. Port Configuration Diagram of Serial Array Unit 0**	(a)
p.524 to 535	Change of **11.3 Registers Controlling Serial Array Unit**	(c)
p.542 to 595	Change of **11.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) Communication**	(c)
p.601	Change of description in **Figure 11-75. SPI Function Timing Diagram**	(a)
p.603 to 656	Change of **11.6 Operation of SPI Function (CSI00, CSI01)**	(c)
p.635	Modification of description in **Figure 11-103. Procedure for Resuming Slave Transmission**	(a)
p.662 to 680	Change of **11.7 Operation of UART Communication**	(c)
p.694 to 702	Change of **11.8 Operation of Simplified I^2C (IIC11, IIC20) Communication**	(c)
CHAPTER 12 ASYNCHRONOUS SERIAL INTERFACE LIN-UART (UARTF)		
p.715	Modification of description in **Figure 12-1. Block Diagram of Asynchronous Serial Interface LIN-UART**	(a)
p.718 to 746	Change of **12.3 Control Registers**	(c)
p.718	Change of description in **Figure 12-3. Format of LIN-UARTn Control Register 0 (UFnCTL0)**	(c)
p.720	Change of **Caution** in **Figure 12-4. Format of LIN-UARTn Option Register 0 (UFnOPT0) (1/3)**	(c)
p.721	Change of description in **Figure 12-4. Format of LIN-UARTn Option Register 0 (UFnOPT0) (2/3)**	(c)
p.735	Change of **Caution** and **Remark** in **Figure 12-9. Format of LIN-UARTn Transmit Data Register (UFnTX)**	(c)
p.736	Change of **Caution** and **Remark** in **Figure 12-10. Format of LIN-UARTn 8-bit Transmit Data Register (UFnTXB)**	(c)
p.739	Change of **Caution** in **Figure 12-13. Format of LIN-UARTn Receive Data Register (UFnRX)**	(c)
p.744	Change of **Caution** in **Figure 12-17. Format of LIN-UARTn Buffer Control Register (UFnBUCTL) (2/2)**	(c)
p.747	Change of **12.4 (2) Reception complete interrupt request signal (INTLRn)**	(c)
p.748	Change of description in **12.4 (3) Transmission interrupt request signal (INTLTn)**	(c)
p.749 to 776	Change of **12.5 Operation**	(c)
p.749	Change of Remark in **12.5.1 Data format**	(c)
p.757	Change of **Note** in **Figure 12-28. LIN Reception Manipulation Outline**	(c)
p.762	Change of description in **Figure 12-33. BF Transmission Processing Flow**	(c)
p.765	Change of description in **12.5.6 BF reception**	(c)
p.766	Change of description in **Figure 12-36. BF Reception Timing Example**	(c)
p.768	Change of description and **Caution** in **12.5.8 Data consistency check**	(c)
p.784 to 799	Change of **12.7 LIN Communication Automatic Baud Rate Mode**	(c)
p.788	Modification of description in **Figure 12-56. LIN Communication Automatic Baud Rate Mode (Response Transmission)**	(a)
p.804 to 810	Change of **12.10 Dedicated Baud Rate Generator**	(c)

Remark "Classification" in the above table classifies revisions as follows.

(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(3/4)

Page	Description	Classification
CHAPTER 13 CAN CONTROLLER		
p.830	Change of description in **Table 13-14. Error Counte**	(c)
p.833	Change of description in **Figure 13-18. Segment Setting**	(c)
p.855	Change of **Figure 13-25. Format of Peripheral Clock Select Register (PCKSEL)**	(c)
p.857	Change of **Caution** in **Figure 13-26. Format of CAN Global Module Control Register (CGMCTRL) (2/2)**	(c)
p.866	Change of **Caution** in **Figure 13-31. Format of CAN Module Control Register (CCTRL) (3/4)**	(c)
p.889	Change of **Caution** in **Figure 13-50. Format of CAN Message Control Register m (CMCTRLm) (2/3) (b) Write**	(c)
p.892	Change of **Figure 13-52. Format of Port Mode Registers 1, 7 (PM1, PM7)**	(c)
p.897	Change of description in **13.9.2 Receive data read**	(c)
p.914	Change of description in **13.11.3 Example of using power saving modes**	(c)
p.952	Change of **Figure 13-85. Forced Shutdown Process**	(c)
CHAPTER 16 INTERRUPT FUNCTIONS		
p.988	Change of **16.2 Interrupt Sources and Configuration**	(c)
p.993 to 1004	Change of **16.3 Registers Controlling Interrupt Functions**	(c)
CHAPTER 17 KEY INTERRUPT FUNCTION		
p.1017, 1018	Change of **17.3 Register Controlling Key Interrupt**	(c)
CHAPTER 18 STANDBY FUNCTION		
p.1019, 1020	Change of **18.1 Standby Function and Configuration**	(c)
CHAPTER 19 RESET FUNCTION		
p.1049	Change of **19.1 Register for Confirming Reset Source**	(c)
CHAPTER 20 POWER-ON-CLEAR CIRCUIT		
p.1051	Change of **20.1 Functions of Power-on-Clear Circuit**	(c)
p.1052	Modification of description in **Figure 20-1. Block Diagram of Power-on-Clear Circuit**	(a)
p.1052	Change of **20.3 Operation of Power-on-Clear Circuit**	(c)
CHAPTER 21 LOW-VOLTAGE DETECTOR		
p.1059	Change of **21.3 (2) Low-voltage detection level select register (LVIS)**	(c)
p.1060	Change of description in **Figure 21-4. Format of Port Mode Register 12 (PM12)**	(a)
p.1067	Change of **21.4 Operation of Low-Voltage Detector**	(c)
p.1079	Change of **21.5 Cautions for Low-Voltage Detector**	(c)
CHAPTER 22 REGULATOR		
p.1080	Change of **22.1 Regulator Overview**	(c)
p.1081	Change of **22.2 Operation**	(c)
CHAPTER 23 OPTION BYTE		
p.1082	Change of **23.1.1 User option byte (000C0H to 000C2H/020C0H to 020C2H)**	(c)
p.1086	Change of **Caution** in **Figure 23-2. Format of Option Byte (000C1H/020C1H)**	(c)
p.1086	Modification of description in **Figure 23-3. Format of Option Byte (000C2H/020C2H)**	(a)
p.1089	Modification of description in **23.4 Setting of Option Byte**	(a)

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

(4/4)

Page	Description	Classification
CHAPTER 24 FLASH MEMORY		
p.1093	Change of **24.1 Overview**	(c)
p.1095	Change of **24.2 Registers that Control Flash Memory**	(c)
p.1096	Change of **24.3 Data Flash Access Procedure**	(c)
p.1101 to 1103	Change of **24.7 Connection of Pins on Board**	(c)
p.1107	Modification of description in **24.9 Security Settings**	(a)
CHAPTER 26 ON-CHIP DEBUG FUNCTION		
p.1124	Modification of description in **26.2 On-Chip Debug Security ID**	(a)
p.1125	Modification of description in **Figure 26-2. Memory Spaces Where Debug Monitor Programs Are Allocated**	(a)
CHAPTER28 INSTRUCTION SET		
p.1131	Change of description in **Table 28-2. Symbols in “Operation” Column**	(c)
p.1133 to 1149	Change of **28.2 Operation List**	(c)
CHAPTER 29 ELECTRICAL SPECIFICATIONS		
p.1152 to 1208	Change of **CHAPTER 29 ELECTRICAL SPECIFICATIONS**	(a)
p.1160	Change of description in **DC Characteristics (5/12)**	(a)
p.1161	Change of description in **DC Characteristics (6/12)**	(a)
p.1165	Change of description in **DC Characteristics (10/12)**	(a)
p.1166	Change of description in **DC Characteristics (11/12)**	(a)
p.1167	Change of description in **Notes** in **DC Characteristics (11/12)**	(c)
p.1171	Change of description in **Interrupt Request Input Timing**	(a)
p.1206	Change of description in **LVI Circuit Timing**	(a)
p.1208	Change of condition in **Data Flash Memory Programming Characteristics**	(a)

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

B.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/4)

Page	Description	Chapter
2nd	Addition of **LVIOUT** to alternate function of P70	Throughout
	Change of name of CCSmm bit of serial mode register mm (SMRmm) to SCCSmm bit	
	Change of **Caution** in **1.4 Pin Configuration (Top View)**	**CHAPTER 1 OUTLINE**
	Addition of **LVIOUT/P70 pin** to **1.6 Block Diagram**	
	Addition of the **CSI (supports SPI)** to **Serial interface** in **1.7 Outline of Functions**	
	Change of **2.2.3 (1)**	**CHAPTER 2 PIN FUNCTIONS**
	Change of **2.2.7 (2) (a) INTP5, INTP6**	
	Change of **Table 2-3. Connection of Unused Pins**	
	Addition of **Figure 2-1. Pin I/O Circuit List**	
	Change of **Table 3-5. SFR List**	**CHAPTER 3 CPU ARCHITECTURE**
	Change of **Table 3-6. Extended SFR (2nd SFR) List**	
	Change of **Table 4-4. Port Configuration**	**CHAPTER 4 PORT FUNCTIONS**
	Change of **Figure 4-47** to **Figure 4-50**	
	Change of description in **4.2.8 Port 8**	
	Change of description in **4.2.9 Port 9**	
	Change of description in **4.2.10 Port 10**	
	Change of **Caution 3** in **Figure 4-92. Format of A/D Port Configuration Register (ADPC)**	
	Change of description in **4.4.4 (2) Setting procedure when using I/O pins of simplified IIC11 function**	
	Change of P140-PCL in **Table 4-8. Settings of Port Mode Register and Output Latch When Using Alternate Function (5/5)**	
	Change of **Figure 5-1. Block Diagram of Clock Generator** and addition of **Note**	**CHAPTER 5 CLOCK GENERATOR**
	Change of **Remark** in **5.3 (3) Oscillation stabilization time counter status register (OSTC)**	
	Change of **After reset** and **Caution** in **5.3 (5) System clock control register (CKC)**	
	Change of **Caution** in **Figure 5-8. Format of Peripheral Enable Registers 0, 1 (PER0, PER1)**	
	Addition of **Note** and **Caution** to **Figure 5-9. Format of Peripheral Clock Select Register (PCKSEL)**	
	Change of **PLLDIV1 bit** in **Figure 5-11. Format of PLL Control Register (PLLCTL)**	
	Change of **address** in **Figure 5-12. Format of PLL Status Register (PLLSTS)**	
	Change of description in **5.4 Clock Monitor**	
	Change of **Figure 5-17. Clock Generator Operation When Power Supply Voltage Is Turned On** and change of **Caution**	
	Change of **Caution** in **5.7.1 (3) Example of setting procedure when using high-speed system clock as CPU/peripheral hardware clock**	
	Change of **Caution 2** in **Figure 6-8. Format of Peripheral Enable Register 0 (PER0)**	**CHAPTER 6 TIMER ARRAY UNIT**
	Change of description of **TIS15 bit** in **Figure 6-28. Format of Timer Input Select Register 1 (TIS1) (1/2)**	
	Change of description of **TOS15 bit** in **Figure 6-30. Format of Timer Output Select Register 1 (TOS1)**	
	Change of **P70 pin** in **Table 6-5. Port Register and Port Mode Register Settings When Using Timer (2/3)** and addition of **Note**	

(2/4)

Page	Description	Chapter
2nd	Change of **Caution** in **Figure 7-2. Format of Peripheral Enable Register 1 (PER1)**	**CHAPTER 7 16-BIT WAKEUP TIMER**
	Addition of **Caution** to **8.3 (2) Register used to change the operating mode of the watchdog timer during self programming (WDTSELF)**	**CHAPTER 8 WATCHDOG TIMER**
	Addition of **Caution 2** in **Figure 10-2. Format of Peripheral Enable Register 0 (PER0)**	**CHAPTER 10 A/D CONVERTER**
	Change of **Figure 10-6. Format of A/D Converter Mode Register 1 (ADM1)**	
	Change of **Caution 3** in **Figure 10-11. Format of A/D Port Configuration Register (ADPC)**	
	Change of **Table 10-2. Number of conversion clocks per channel (No Discharge)**	
	Change of **Table 10-3. Number of conversion clocks per channel (Discharge Performed)**	
	Change of **Table 10-4. A/D Conversion Time by CPU/Peripheral Hardware Clock Frequency (High-Speed Mode 1 : ADHS1, ADHS0 = 0, 0) (No Discharge)**	
	Change of **Table 10-9. A/D Conversion Time by CPU/Peripheral Hardware Clock Frequency (Normal Mode : ADHS1, ADHS0 = 1, 0) (Discharge Performed)**	
	Change of description in **10.4.3 Trigger mode selection**	
	Change of description in **10.6 (4) Conflicting operations**	
	Change of **Table 10-11. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)**	
	Change of **Caution 3** in **Figure 11-6. Format of Peripheral Enable Registers 0, 1 (PER0, PER1)**	**CHAPTER 11 SERIAL ARRAY UNIT**
	Change of **Figure 11-10. Format of Serial Data Register mn (SDRmn)**	
	Change of **Caution** in **Figure 11-19. Format of Serial Slave Select Enable Register (SSE0)**	
	Change of description in **11.3 (16) Noise filter enable register 0 (NFEN0)**	
	Change of **Figure 11-24. Format of Port Mode Registers 1, 3, 4, 6, 7, and 15 (PM1, PM3, PM4, PM6, PM7, PM15)**	
	Change of **Caution 2** in **Figure 11-25. Peripheral Enable Registers 0, 1 (PER0, PER1) Setting When Stopping the Operation by Units**	
	Change of **Table 11-8** to **Table 11-12**	
	Addition of **Caution** to **Figure 12-2. Format of Peripheral Enable Register 0 (PER0)**	**CHAPTER 12 ASYNCHRONOUS SERIAL INTERFACE LIN-UART (UARTF)**
	Change of **Table 12-2. Interrupts and Their Default Priorities**	
	Change of description in **12.11 Cautions for Use**	
	Change of address in **Figure 13-31. Format of CAN Module Control Register (CCTRL)**	**CHAPTER 13 CAN CONTROLLER**
	Change of **13.9.5 Multi buffer receive block function**	
	Change of **Figure 14-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)**	**CHAPTER 14 MULTIPLIER/DIVIDER**
	Change of register name and bit name of **DMA all-channel forced wait register**	**CHAPTER 15 DMA CONTROLLER**

(3/4)

Page	Description	Chapter
2nd	Change of **Table 16-1. Interrupt Source List**	**CHAPTER 16 INTERRUPT FUNCTIONS**
	Change of **16.3 Registers Controlling Interrupt Functions**	
	Addition of **Caution** to **Figure 16-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (78K0R/HG3)**	
	Addition of **Caution** to **Figure 16-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) (78K0R/HG3)**	
	Addition of **Caution** to **Figure 16-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H(78K0R/HG3))**	
	Change of **Table 16-3. Ports Corresponding to EGPn and EGNn**	
	Change of **Remark** in **18.1.2 (2) Oscillation stabilization time counter status register (OSTC)**	**CHAPTER 18 STANDBY FUNCTION**
	Change of **Figure 18-6. HALT Mode Release by Reset**	
	Change of **Figure 18-9. STOP Mode Release by Reset**	
	Change of **Figure 19-2. Timing of Reset by $\overline{\text{RESET}}$ Input**	**CHAPTER 19 RESET FUNCTION**
	Change of **Figure 19-3. Timing of Reset Due to Execution of Illegal Instruction or Watchdog Timer Overflow**	
	Change of **System clock control register (CKC)** in **Table 19-2. Hardware Statuses After Reset Acknowledgment**	
	Change of **Figure 20-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1) When LVI is OFF upon power application (option byte: LVIOFF = 1)**	**CHAPTER 20 POWER-ON-CLEAR CIRCUIT**
	Change of **Figure 20-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2) When LVI is ON upon power application (option byte: LVIOFF = 0)**	
	Change of **21.4.1 (1) When detecting level of supply voltage (V_{DD})**	**CHAPTER 21 LOW-VOLTAGE DETECTOR**
	Change of **21.4.1 (2) When detecting level of input voltage from external input pin (EXLVI)**	
	Addition of **Caution** to **24.1.2 Data flash memory features**	**CHAPTER 24 FLASH MEMORY**
	Addition of **Caution** to **Figure 24-1. Format of Peripheral Enable Registers 1 (PER1)**	
	Addition of **Caution** to **24.10 Code Flash Memory Programming by Self-Programming**	
	Change of description in **25.1.1 (1) Illegal-memory access detection control register (IAWCTL)**	**CHAPTER 25 DATA PROTECTION AND SAFETY**
	Change of **25.2 (1) Specific-register manipulation protection register (GUARD)**	
	Addition of **Note** to **Figure 26-1. Connection Example of QB-MINI2 and 78K0R/Hx3**	**CHAPTER 26 ON-CHIP DEBUG FUNCTION**
	Change of **28.2 Operation List**	**CHAPTER 28 INSTRUCTION SET**

(4/4)

Page	Description	Chapter
2nd	Addition of **PLL Circuit Characteristics**	**CHAPTER 29 ELECTRICAL SPECIFICATIONS**
	Change of **Input leakage current, high (I_{LIH4})** in **DC Characteristics**	
	Change of condition of **On-chip pll-up resistance** in **DC Characteristics**	
	Change of **Supply current** in **DC Characteristic**	
	Change of **Data flash operating current** in **DC Characteristic**	
	Change of **TOmn output frequency** in **AC Characteristics (1) Basic operation**	
	Change of Output rise time, fall time in **AC Characteristics (1) Basic operation**	
	Addition of **Note** to **AC Timing Test Points** in **AC Characteristics (1) Basic operation**	
	Addition of Output Rise Time, Fall Time Timing to **AC Characteristics (1) Basic operation**	
	Change of **(2) Serial interface: Serial array unit**	
	Change of **Data Flash Memory Programming Characteristics**	
	Change of Part Number in **A.5.1 When using in-circuit emulator QB-78K0RFX3**	**APPENDIX A DEVELOPMENT TOOLS**

78K0R/Hx3 User’s Manual: Hardware

Publication Date:	Rev.2.00	Sep 16, 2011
	Rev.3.00	Sep 25, 2013
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: +44-1628-651-700, Fax: +44-1628-651-804

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.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 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 1.3

78K0R/Hx3

Renesas Electronics Corporation